Mikko Pänkäälä

Potential and Challenges of Analog Reconfigurable Computation in Modern and Future CMOS

Turku Centre *for* Computer Science

TUCS Dissertations
No 187, December 2014

Potential and Challenges of Analog Reconfigurable Computation in Modern and Future CMOS

Mikko Pänkäälä

To be presented, with the permission of the Faculty of Mathematics and Natural Sciences of the University of Turku, for public criticism in Auditorium Beta on December 5, 2014, at 12 noon.

University of Turku
Department of Information Technology &
Technology Research Center (TRC)
20014 Turun Yliopisto

2014

Supervisors

Adjunct Professor Mika Laiho
Technology Research Center
University of Turku
FIN-20014 University of Turku
Finland

Dr. Tech. Jonne Poikonen
Technology Research Center
University of Turku
FIN-20014 University of Turku
Finland

Reviewers

Dr. Christopher M. Twigg
Department of Electrical and Computer Engineering
Binghamton State University of New York
PO Box 6000
Binghamton, NY 13902-6000
United States of America

Victor Manuel Brea Sánchez
Department of Electronics and Computer Science
University of Santiago de Compostela
Edificio Monte da Condesa, Campus Vida
15782 Santiago de Compostela
Spain

Opponent

Associate Professor Ricardo Carmona-Galán
Institute of Microelectronics of Seville (IMSE-CNM)
Spanish National Research Council (CSIC)
Parque Científico y Tecnológico Cartuja, Calle Américo Vespucio s/n
41092 Sevilla
Spain

ISBN 978-952-12-3141-4
ISSN 1239-1883

Abstract

In this work, the feasibility of the floating-gate technology in analog computing platforms in a scaled down general-purpose CMOS technology is considered. When the technology is scaled down the performance of analog circuits tends to get worse because the process parameters are optimized for digital transistors and the scaling involves the reduction of supply voltages. Generally, the challenge in analog circuit design is that all salient design metrics such as power, area, bandwidth and accuracy are interrelated. Furthermore, poor flexibility, i.e. lack of reconfigurability, the reuse of IP etc., can be considered the most severe weakness of analog hardware. On this account, digital calibration schemes are often required for improved performance or yield enhancement, whereas high flexibility/reconfigurability can not be easily achieved. Here, it is discussed whether it is possible to work around these obstacles by using floating-gate transistors (FGTs), and analyze problems associated with the practical implementation. FGT technology is attractive because it is electrically programmable and also features a charge-based built-in non-volatile memory. Apart from being ideal for canceling the circuit non-idealities due to process variations, the FGTs can also be used as computational or adaptive elements in analog circuits.

The nominal gate oxide thickness in the deep sub-micron (DSM) processes is too thin to support robust charge retention and consequently the FGT becomes leaky. In principle, non-leaky FGTs can be implemented in a scaled down process without any special masks by using "double"-oxide transistors intended for providing devices that operate with higher supply voltages than general purpose devices. However, in practice the technology scaling poses several challenges which are addressed in this thesis.

To provide a sufficiently wide-ranging survey, six prototype chips with varying complexity were implemented in four different DSM process nodes and investigated from this perspective. The focus is on non-leaky FGTs, but the presented autozeroing floating-gate amplifier (AFGA) demonstrates that leaky FGTs may also find a use. The simplest test structures contain only a few transistors, whereas the most complex experimental chip is an implementation of a spiking neural network (SNN) which comprises thousands of active and passive devices. More precisely, it is a fully connected (256 FGT synapses) two-layer spiking neural network (SNN), where the adaptive properties of FGT are taken advantage of. A compact realization of Spike Timing Dependent Plasticity (STDP) within the SNN is one of the key contributions of this thesis.

Finally, the considerations in this thesis extend beyond CMOS to emerging nanodevices. To this end, one promising emerging nanoscale circuit element - memristor - is reviewed and its applicability for analog processing is considered. Furthermore, it is discussed how the FGT technology can be used to prototype computation paradigms compatible with these emerging two-terminal nanoscale devices in a mature and widely available CMOS technology.

Tiivistelmä

Tässä työssä tarkastellaan kelluvahilaiseen transistoriin (FGT, Floating-Gate Transistor) pohjautuvan analogisen laskenta-alustan toteutettavuutta yleiskäyttöisillä erittäin pienen viivanleveyden CMOS-teknologioilla. Tyypillisesti analogiapiirien suorituskyky heikkenee, kun teknologiaa skaalataan pienemmäksi, koska skaalaukseen liittyy käyttöjännitteiden alentaminen. Lisäksi prosessiparametrit optimoidaan yleensä digitaalipiireille. Yleisesti ottaen analogiasuunnittelun haasteena on, että kaikki oleelliset suunnittelussa käytetyt mittarit, kuten teho, pinta-ala, kaistanleveys ja tarkkuus riippuvat toisistaan. Analogiapiirien heikkouksiksi voidaan lisäksi lukea heikko muunneltavuus, eli suunnitellut piirit eivät välttämättä sovellu suoraan uudelleen käytettäviksi yms. Tämän vuoksi analogiapiirien toimintaa korjataan usein digitaalisilla kalibrointiratkaisuilla suorituskyvyn ja/tai saannon parantamiseksi, kun taas analogiapiirien hyvä muunneltavuus on vaikeampi toteuttaa. Tässä työssä tarkastellaan voidaanko edellä mainitut analogiapiirien heikkoudet kiertää käyttämällä toteutuksessa kelluvahilaisia transistoreita ja analysoidaan käytännön toteutukseen liittyviä ongelmia. FGT teknologia on houkutteleva, koska sitä ohjelmoidaan sähköisesti ja siihen on sisäänrakennettu sähkövaraukseen perustuva haihtumaton muisti. FGT teknologia on ihanteellinen ratkaisu prosessivaihteluista johtuvien epäideaalisuuksien kumoamiseen, mutta sitä voidaan käyttää myös laskenta- tai mukautuvana elementtinä analogiapiireissä.

Kun mennään erittäin pienen viivanleveyden ($<< 1\ \mu$m) prosesseihin, hilaoksidin nimellispaksuus ohenee niin paljon, että kelluvahilainen transistori alkaa vuotaa, eikä se enää kykene luotettavasti pitämään sähkövarausta muistissa. Periaatteessa vuotamattoman kelluvahilaisen transistorin toteuttaminen on suoraviivaista myös erittäin pienen viivanleveyden prosesseissa käyttämällä paksun hilaoksidin transistoria. Paksun hilaoksidin transistori on tarkoitettu sellaisten piirikomponenttien toteuttamiseen, joiden jännitekestoisuuden tulee olla parempi kuin yleiskäyttöisten piirikomponenttien. Käytännössä teknologian skaalaus aiheuttaa kuitenkin useita haasteita, joita käsitellään tässä työssä.

Riittävän kattavan otoksen saamiseksi skaalauksesta aiheutuvien ongelmien tarkasteluun, tätä tutkimusta varten toteutettiin kaikkiaan kuusi erilaista, kompleksisuudeltaan vaihtelevaa, prototyyppisirua käyttäen neljää eri erittäin pienen viivanleveyden prosessia. Tarkastelun painopiste on vuotamattomissa kelluvahilaisissa transistoreissa, mutta itsestään nollaavan FGT-vahvistimen (AFGA, Autozeroing Floating-gate Amplifier) avulla demonstroidaan miten myös vuotavia kelluvahilaisia transistoreita voidaan hyödyntää. Yksinkertaisimmat testirakenteet sisältävät vain muutaman transistorin, kun taas monimutkaisin siru on toteutus neuroverkosta, joka koos-

tuu tuhansista aktiivisista ja passiivisista piirielementeistä. Tarkemmin sanottuna toteutus on täysin kytketty (256 FGT synapsia) kaksikerroksinen virtasykäyksien avulla kommunikoiva neuroverkko (SNN, Spiking Neural Network), jossa hyödynnetään kelluvahilaisen transistorin mukautumisominaisuutta. Virtasykäyksien avulla kommunikoivan neuroverkon kompakti toteutus, jossa synapsien muovautuvuus riippuu neuronien lähettämien virtasykäyksien keskinäisistä ajoituksista (STDP, Spike Timing Dependent Plasticity), on tämän väitöskirjan yksi keskeisimpiä saavutuksia.

Väitöskirjan alkuosa käsittelee lähinnä CMOS teknologiaa, mutta kirjan lopussa näkökulmaa laajennetaan koskemaan myös uusia nanoskaalan piirielementtejä. Lähempään tarkasteluun otetaan yksi lupaava nanoskaalan piirielementti - memristori - jonka soveltuvuutta analogiaprosessointiin arvioidaan. Lisäksi tarkastellaan miten FGT-teknologian avulla voidaan testata laskentaparadigmoja, jotka ovat yhteensopivia näiden uusien kaksiporttisten nanoskaalan piirielementtien kanssa. Tämän lähestymistavan etuna on, että toteutuksessa voidaan käyttää kypsää ja laajasti saatavilla olevaa CMOS-teknologiaa.

Acknowledgements

I wish to thank the University of Turku for providing me an interesting place to work and Turku Centre for Computer Science (TUCS) Graduate School for the support.

Professor Ari Paasio, you believed in me, accepted me as a postgraduate student and opened the door to the interesting world of science for me, so without you there would be no this book. I am truly grateful to Dr. Christopher M. Twigg and Dr. Victor M. Brea Sánchez for kindly reviewing this thesis. Especially Dr. Twigg examined the manuscript rigorously and gave very valuable critique which eventually led to significant improvements in the quality of the manuscript. I'm also very grateful to my supervisors Dr. Mika Laiho and Dr. Jonne Poikonen for generously contributing their time, reading my dissertation, providing excellent comments and suggestions, helping me with the measurements and particularly getting me past setbacks experienced in the finishing phase of the work.

I would like to thank all my current colleagues at the TRC and my former colleagues at the IT Department for their encouragement, support, assistance, influence and friendship. I have been privileged to work with many talented, really smart and exceptional individuals over these years. I have spent many long hours at the lab together with some of you desperately trying to finish the chip designs in time for the process runs. I bet that I will never forget that, nor cease being surprised that despite the hurry and tiredness most of the designed chips operate without reproach. Special thanks to Sami Nuuttila and Peter Virta for their patient assistance with the computers in spite of my obsession to always try to persuade the circuit design tools to do things no one in our lab has never tried before. In addition, Peter has done great job in preparing the required PCBs for the chip evaluations.

Last but not least, I want to express my sincerest gratitude to my ever loving wife, Katri, who has had to endure many things throughout this process and still has been with me every step of the way.

Contents

List Of Acronyms

ADC	Analog-to-Digital-Converter
AER	Address-Event-Representation
AFGA	Autozeroing Floating-gate Amplifier
ANN	Artificial Neural Network
AR	Auger Recombination
ASIC	Application Specific Integrated Circuit
ASP	Analog Signal Processing/Processor
CHE	Channel Hot-Electron
CHH	Channel Hot-Hole
CMOL	CMOS/Molecular Hybrid
CMOS	Complementary Metal Oxide Semiconductor
DAHC	Drain-Avalanche Hot-carrier
DC	Direct Current
DCT	Discrete Cosine Transform
DRC	Design rule Check
DSM	Deep sub-micron
DSP	Digital Signal Processor
DST	Discrete Sine Transform
ESD	Electrostatic Discharge
FG	Floating-gate
FGT	Floating-gate Transistor
F-N tunneling	Fowler Nordheim tunneling
FPAA	Field Programmable Analog Array
FPGA	Field Programmable Gate Array
FPNI	Field Programmable Nanowire Interconnect
I-F	Integrate-and-Fire
IIHEI	Impact Ionized Hot-Electron Injection
IO	Input Output
LFBGA	Low profile Fine pitch Ball Grid Array
MAC	Multiply Accumulate operation
MITE	Multi Input Translinear Element

MIM-capacitor	Metal-Insulator-Metal-capacitor
MOSFET	Metal Oxide Semiconductor Field Effect Transistor
MRAM	Magnetoresistive Random Access Memory
MUX	Multiplexer
NDA	Non-disclosure Agreement
OpAmp	Operational Amplifier
PDK	Process Design Kit
PRAM	Phase Change Random Access Memory
RF	Radio Frequency
SC	Switched Capacitor
SGHE	Secondarily Generated Hot-Electron
SHE	Substrate Hot-Electron
SNN	Spiking Neural Network
SNR	Signal-to-noise ratio
SoC	System on a Chip
STDP	Spike Timing Dependent Plasticity
TDDB	Time Dependent Dielectric Breakdown
UV	Ultraviolet
VLSI	Very Large Scale Integration

Chapter 1

Introduction

1.1 The Role of Analog Circuits in Signal Processing

In today's modern affluent society, we are surrounded by a myriad of electronic devices such as digital cameras, mp3 players, cellular phones, laptops and digital TVs. Ubiquitous digital gadgets may easily mislead one to conclude that analog circuits have completely vanished. This illusion stems from labeling a product with a "digital" prefix, e.g. a digital camera, which simply means that it has a digital signal processor (DSP) and information is stored in a digital format. The word digital is often used as a marketing argument, to highlight the efficiency and ease of use of the product. However, a good commercial product can rarely be made with a successful digital design only. A "digital" product often includes a surprising number of analog circuits. For example, DSP manufacturer Texas Instruments stated that for every DSP in an electronic system, there are approximately ten analog components [1]. Indeed, analog circuits are needed as an interface between digital processing and the external world. In this context, the interface circuitry covers tasks like transforming the sensed quantity (for example photons or acoustic waves) into a useful electronic magnitude (for example voltage, current, charge or frequency). Other analog tasks include filtering, data conversions, and transforming an electronic signal back to perceivable signal (for example movement, sound or light intensity). In that sense, analog components are an essential and irreplaceable part of the "digital" systems.

It is not absolutely necessary to visit the digital domain between the analog input and the analog output. However, in practice, this is usually the case at least if more or less a "general purpose" signal processor is considered. This is simply because digital hardware is more flexible than analog hardware. Due to the better flexibility, the implementation of algorithms is often more straightforward, and for example precision can be adjusted according to the requirements. For some applications, a Field Programmable Gate Array (FPGA) or an Application Specific Integrated Circuit (ASIC) may be a better solution than a DSP. The choice between different processing platforms depends on specifications like power budget and die area constraints. Other factors include the processing architecture (for example serial or parallel) and resources (time and money) that are devoted in the project. Whatever the case, programmability, scalability over different technology nodes, computational accuracy and immunity to noise are properties hard to achieve with analog design. Consequently, Analog Signal Processors (ASPs) are rarely seen on commercial systems, but DSPs are the de facto industry standard. However, in some special cases it can be beneficial or even necessary to rely on ASP instead of DSP. For example, in very high-speed or in very low-power applications the extra data conversions (analog to digital and then back to analog) can be too expensive in terms of delay or power consumption. Nevertheless, it is very unlikely that ASP would widely replace DSPs in the near future, but rather complement them.

Programmable analog signal processors have been successfully implemented, for example the Field Programmable Analog Array (FPAA) [2]. FPAAs have their disadvantages and limitations, as explained in later chapters, which prevents their use as a general purpose computing platform in complicated systems. However, analog circuits have dominated in the implementation of many types of neural networks, because many of the algorithms and functions used in this field map nicely and efficiently to analog hardware. Data processing in neural networks is also organized so that it works even if the elementary processing elements have high noise and low accuracy, which is common to low power, small area analog circuits.

1.2 Programmable Analog Technologies

An implementation of an analog Discrete-Cosine-Transform-processor (DCT) was previously studied by the author [3, 4]. In this work, the accuracy of the transistors was identified to be the most serious roadblock for the implementation of a DCT-based motion estimation algorithm. Accuracy enhancement can be achieved simply by using large devices at the cost of silicon area and increased power consumption due to a larger capacitive load.

An alternative to spending more area is to fine-tune the transistors after fabrication for better accuracy. This can be realized by offloading the accuracy constraints to a digital processor, a technique which is commonly known as “digitally assisted analog circuits”. The feasibility of this design technique improves with technology scaling, but needs some hardware overhead for data processing and storing the configuration bits (ideally to a non-volatile digital memory). As an example this technique has been applied to the performance enhancement of Analog-to-Digital-Converters (ADC) and Digital-to-Analog-Converters (DAC) [5].

On the other hand in [3, 4], reconfigurability was also identified to be an important feature that greatly improves the reuse of designed analog circuit blocks. For example, besides DCT the same analog circuit could be used for implementing the Discrete-Sine-Transform (DST) if the coefficients were programmable. Reconfigurability was seen as such an important feature that it eventually led to the exploration of programmable analog technologies.

There are a few technologies for adding programmability to the analog hardware. In this context, programmability means the ability to repeatedly reconfigure the design to implement different functionality and / or to improve the performance by changing the operation point of the transistor(s). This definition excludes traditional post fabrication programming methods like laser trimming, which can be done only once. Post fabrication methods are mainly used for example to cancel mismatches for improved accuracy, or to enhance yield in general. Over the years, there have been many attempts to realize an analog counterpart to the FPGA. Many of these attempts are based on switched Capacitors (SC) accompanied with operational amplifiers (OpAmp). Another technology providing means to achieve reconfigurable analog hardware is the Floating-Gate Transistor (FGT).

The FGT technology provides the basis for the billion dollar flash-memory business, a memory type that is widely used for storing information in any application where compact size and non-volatility are appreciated. The programming of FGTs involves manipulating the charge stored on the floating-gate. This manipulation is effectively done by utilizing quantum mechanical mechanisms like electron tunneling and hot electron injection. In fact, FGT technology is currently in practice the only viable alternative to implement a non-volatile memory in CMOS. Besides digital memory applications, the FGTs can also act as programmable analog computational elements (with intrinsic analog memory feature). In practice, this extension to analog applications is achieved simply by increasing the programming resolution of the FGT from two separate states to a continuum of states. For example, the above-mentioned FPAAs are based on floating-gate technology. FGTs allow for more versatile and compact designs as compared to SC-OpAmp designs. However, there are fundamental difficulties in realizing a truly versatile high performance FPAA, such as the granularity of the building blocks and the influence of routing on performance.

A more recent technology for nonvolatile storage is the memory resistor, or memristor for short, which also enables reconfigurable designs. Although, the theoretical model for the memristor was introduced already in 1971 [6], the physical implementation of this component was successfully linked to the memristor theory just recently [7]. The article in Nature drew a lot of attention and got more researchers involved. As a result, plenty of academic resources were rapidly devoted to investigating new architectures and applications for the memristor.

Besides memristor, there are other emerging devices rushing to market like Phase Change Random Access Memory (PRAM) [8] and Magnetoresistive Random Access Memory (MRAM) [9] which are capable for storing (non-volatile) their current state. These emerging devices are primarily seen as candidates for non-volatile digital memory applications which can potentially replace the dominant flash memory in the future. However, some of these devices may serve as programmable analog computational elements as well, likewise FGT. There are already many proposals in the literature how to use memristors for analog computation. Most of these new devices are two-terminal thin film structures and their operation principle is essentially based on manipulating the resistivity of the material. From this follows, that their usage as computational elements requires a paradigm shift from the traditional computation architectures which are based on three-terminal (four-terminal, if the bulk terminal is counted in) transistors.

This thesis aims to explore the limitations and opportunities on realizing programmable analog hardware in CMOS, focusing on FGT technology. However, since FGTs share many essential characteristics with memristors, also many design aspects are similar. Furthermore, it is possible to configure an FGT device to be virtually a two-terminal device the channel conductance of which can be electrically manipulated. Therefore all future two-terminal devices based on resistivity change can be mimicked with FGT devices and consequently the CMOS-based FGT technology allows the exploration of computation architectures which are compatible with these emerging devices. To this end, as a case study, a compact realization of Spike Timing Dependent Plasticity (STDP is a biological learning method) is proposed and implemented on CMOS-based FGTs that are configured as two-terminal memory devices. Therefore, in this respect the considerations in this thesis extend beyond CMOS, to memristor-based analog hardware. It is likely that memristive memories will be available as CMOS add-ons.

1.3 Problem Description and Scope

There are many research groups who are actively using FGTs in their analog circuits. Most of the research related to the usage of FGTs in analog design is based on the seminal research conducted by Caltech professor Carver Mead, whose students have continued to make good use of the properties of FGTs and gained significant progress for example with the programming framework and extending the application domain.

FGTs are not typically included in the characterization process provided by the foundry. For a previously uncharacterized process, certain empirical parameters are unknown, as is the case with the examined processes in this thesis. Therefore, before one can concentrate on designing effective and reliable solution for a given problem, taking full advantage of the properties of FGTs, one should characterize the accessible CMOS process kit very well. The characterization is challenging with an academic budget, as taking into consideration that prototyping especially with deep sub-micron technologies is time-consuming and expensive. As devoting one process run just for device characterization in each new technology is not practical, the emphasis in this study is on experimenting with floating-gate devices without first trying to characterize the process kit. Consequently, the focus is on solving low-level problems rather than developing high performance circuits. In practice the most important goal throughout this study has been to have functional FGTs in general. The design is based on intuition and information extracted by interpolation from other processes and studies. From this point of view, the most relevant questions are: What kind of problems will be faced when designing FGT-based circuits using modern sub-micron processes and how to act in the absence of empirical parameters?

A typical System on a Chip (SoC) includes many analog components that must be placed on the same chip due to integration benefits. Despite this, modern general purpose CMOS processes are designed and optimized to support mainly digital designs. The effort to increase the performance of digital hardware is the strongest technological driver and the performance of the analog hardware is secondary. On this account, CMOS analog computation is facing increasingly difficult problems the deeper one proceeds into sub-micron technologies [11].

One of the most essential and fundamental properties of the FGT is the non-volatility of the charge stored on the floating-gate. A metric that is used to evaluate the non-volatility is the retention time. Memory cell's ability to retain charge is dictated by the oxide thickness. That is, only thick enough gate oxide guarantees non-volatility. The nominal gate oxide has become thinner when the CMOS processing technology has evolved toward smaller line widths. This is the main reason (in addition to cheaper manufacturing costs) for using a relatively old technology to implement FGTs.

Retention time that is considered to non-volatile operation is ambiguous in analog memory applications: retention time must be coupled to the allowed amount of error over the examined time period to provide meaningful measure when analog FGT memory cells are considered. However, non-volatility is a term that is frequently used in the literature. In general, the non-volatility and the long-term charge retention are associated to time scales supported by the natural gate oxide thickness roughly down to 0.25 micron technology node. Below this technology node the gate leakage current grows rapidly and the retention times decrease accordingly. In this case the term quasi-/pseudo- or semi-floating-gate transistor is often referred to.

In general, research concentrating on non-volatile FGTs is typically implemented with 0.25 micron or older processing technology, whereas pseudo-FGTs are implemented with 0.18 or newer technology. It is also possible to use denser than 0.25 micron technology to implement non-leaky FGTs, since many foundries provide possibility to use thick oxide devices along with thin oxide devices within certain manufacturing rules. The implementation of (mainly) non-leaky FGTs in a modern deep sub-micron general purpose CMOS process is discussed in this thesis. Various aspects of the restrictions posed by the processing technology itself are discussed. For example, certain amount of energy is required in order to manipulate the charge stored on the floating node of the FGT, so that, the thicker the oxide the more energy is required. To achieve a sufficient energy level during the charge manipulation, the gate oxide of the non-leaky FGT must be exposed to harsh electrical stress - a situation that is usually avoided with regular transistors to prevent transistors from breaking down prematurely. For electron tunneling and injection it is necessary to use voltage levels far beyond the nominal supply rails. Either off-chip or on-chip voltage sources can be used to provide the required high voltages. However, an intention to use an off-chip voltage source is typically poorly supported by the IO-cell library provided by the foundry. Also, it can be challenging to implement high-voltage tolerant switches in a given technology for internal routing of the high voltages to a desired target. This thesis provides practical solutions as to how one can try to solve such issues.

1.4 Structure and Contributions of the Thesis

This thesis is organized into eight chapters. Chapter 2 provides the reader background information, definitions, terminology and concepts related to the floating-gate transistors frequently used in later chapters of this thesis. Chapters 3, 4 and 6 are devoted to considering the actual problem description. Chapter 3 focuses on discussing the compatibility issues of FGTs with the deep sub-micron technologies in general whereas in chapters 4 and 6 the approach is more application oriented. Chapter 4 deals with FGT circuits which are relatively simple, whereas the prototype chip studied in Chapter 6 is much more complex. Namely, the introduced chip is a complete learning system, where FGTs have a key role as they operate as synapses in a Spiking Neural Network (SNN). Chapter 5 provides a brief introduction to biological and artificial neural networks. It is included into this thesis for sake of consistency and to equip the reader with sufficient preliminary knowledge with respect to Chapter 6. The conducted application oriented study is supported by the experimental data collected from various prototype chips manufactured with different process nodes. More precisely, experimental tests comprise the following technology nodes:

- 0.18 μm, one test chip
- 0.13 μm, two test chips
- 90 nm, one test chip
- 65 nm, two test chips

All test chips were implemented with the processes provided by the same foundry. That is, the study offers consistent and comparable data between different process nodes from one foundry. The complexity of the test structures range from a single FGT with all terminals directly connected to respective IO-pads to two layer neural network with 16 neurons in both layers, 256 synapses with interface and auxiliary circuits. Unfortunately, all test chips were not fully functional due to different reasons so that all potentially available measurement data could not be extracted. Extracted measurement data per test chip varies also depending on the implemented test structure itself and applied programming method. Obtained measurement data is compared with simulated data.

In chapter 7, the focus is on emerging devices and their linkage to FGT technology. The discussion concentrates particularly on memristor devices which could potentially replace FGTs in many applications in the future. For the sake of consistency, memristor's potential to replace flash memory and penetration to other digital applications or signal processing in general is also briefly considered.

Finally, the most important findings made in this thesis with associated conclusions are summarized in the last chapter. This monograph contains mostly previously unpublished research results. Due to the focus of this thesis, the individual prototype chips presented in this thesis are mainly very simple and do not implement complex signal processing or similar tasks. That is the reason why only part of the prototype chips is reported outside this thesis. However, all information obtained from experimental prototype chips has been invaluable in creating an overall picture of the feasibility of implementing FGTs in deep sub-micron processes.

Chapter 2

Floating-Gate Transistor

A MOSFET with an extra electrical insulation (capacitor) forms a double gate (gates are in series) device which is called a floating-gate transistor. The first experiments on a MOSFET with an insulated gate date back to 1960 [12]. Since then, floating-gate technology has evolved along two different paths. One path has concentrated on how to use FGTs for binary storage. The other path has investigated how to use FGTs as a computational element in analog circuits. Either way, the charge modification requires electrons to be transported through an insulator (normally $\mathrm{SiO_2}$), which is quite a tricky task and necessitates that certain circumstances are met depending on the method to be used. The performance optimization of an FGT based binary memory has led to the adoption of specialized process options which enables very dense non-volatile binary memory arrays. Currently, there are no process options available which would have been specially designed for implementing analog FGTs, but some process features like double poly layers facilitate the utilization of certain programming techniques. In this thesis, only single poly processes are considered, which means that the discussion is more general and independent of this process option.

The non-disclosure Agreement (NDA) with the process foundry prevents revealing the non-public technological details related to the manufacturing process. Due to this, the evaluation of numerical values is not possible for all equations presented in this thesis. One essential parameter related to FGT technology is the thickness of the gate oxide layer t_{ox}. Nominal t_{ox} of thick oxide devices is less than 6 nm for all technology nodes discussed in this thesis.

2.1 Basic Structure of the FGT

A floating-gate transistor is a device with a capacitively coupled gate (see Fig. 2.1). In other words, the gate has no direct connections to any other

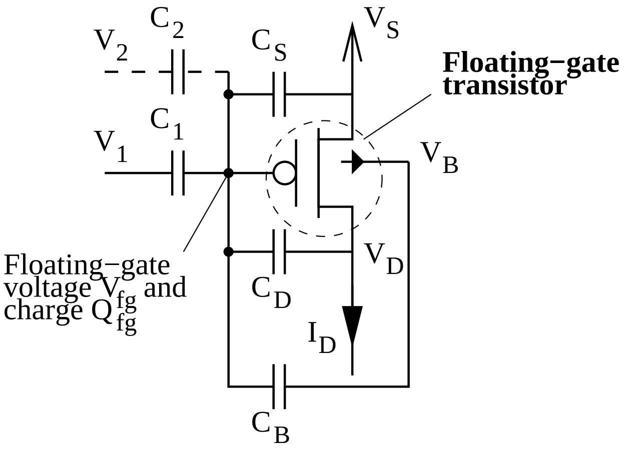


Figure 2.1: Simplified schematic of a single p-type FGT. Charge (Q_{fg}) stored on the floating-gate defines the threshold voltage of the FGT. C_1 and C_2 are capacitive inputs and C_D, C_S and C_B are internal capacitances.

conductors, but is connected to at least one capacitor. Because the actual gate of the transistor is floating, the extra capacitors are usually referred to as control gates. It is instructive to think of this circuit as a basic transconductance amplifier with a capacitive divider in its gate. The capacitive division, κ, is defined as $\frac{C1}{C_{tot}}$, where $C1$ is the input capacitance and C_{tot} the total capacitance connected to the floating-gate, respectively.

It is possible to change the potential on the floating-gate either by capacitively controlled signals or by changing (programming) the charge stored on the gate capacitance. Charge modification is only possible by moving charge carriers (electrons or holes) through the gate insulator. The programming effectively changes the threshold voltage V_{th} of the transistor (see Eq. 2.1 and Fig. 2.2), thus providing means to change the biasing of the transistor, which in turn enables FGT to be used as a computational element. The effective change of V_{th} from the perspective of the control gate is given by

$$\Delta V_{th} = \pm \frac{\Delta Q_{fg}}{C_{tot}} \tag{2.1}$$

where V_{th} is the threshold voltage of the FGT and Q_{fg} is the charge at the floating-gate. The sign of equation is $-$ for the n-type FGT and $+$ for the p-type FGT respectively [1].

The quality and the thickness of the insulators surrounding the floating-gate define the leakage rate of the charge, i.e. the retention time. Even if a memory element is considered non-volatile, the retention time (powered

[1] Here, the charge Q_{fg} is assumed to be positive.

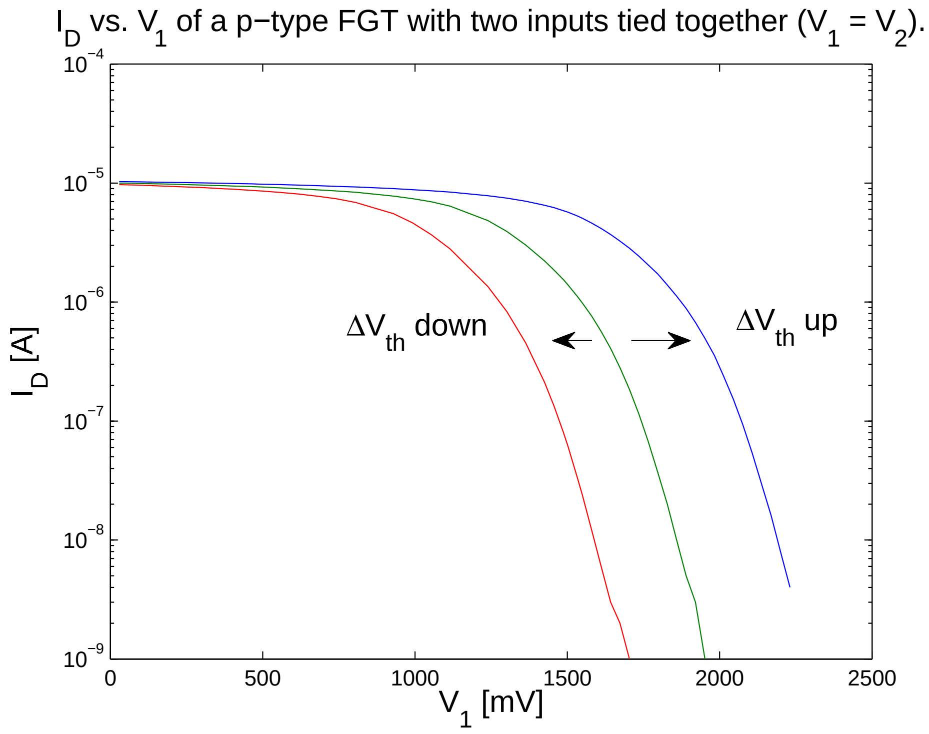


Figure 2.2: Measurement data with a similar setup to that of Fig. 2.1 showing how the programming changes the effective threshold voltage V_{th} of the FGT and how the shift reflects to the I_D vs. V_{GS} curve. Data obtained from a prototype chip manufactured in a 90 nm CMOS process. FGTs are made of thick oxide transistors.

or unpowered FGT) is always finite because of the finite resistivity of the insulator (gate oxide). The retention time dt can be derived from Eq. 2.1.

$$i_{Gtot}dt = \pm C_{tot}dV_{th} \tag{2.2}$$

where i_{Gtot} is the sum of gate currents (leakage). From Eq. 2.2 it becomes apparent that the retention time is related to some allowed value dV_{th} for the threshold voltage shift. Ideally, the preset charge remains nearly unchanged even for years if the power supplies are shut down, hence the name non-volatile memory.

The number of capacitors connected to the floating-gate is dependent on the chosen programming scheme as well as the number of inputs connected to the gate. In general, an FGT has only one actual input, but typically the programming requires connecting more than one capacitor to the floating-gate, therefore it is more useful to mathematically examine the multi-input FGT. By connecting more input capacitors to the floating-gate the result is a Multiple Input Translinear Element (MITE) [13]. Effectively, a MITE produces a drain current that is related to a weighted sum of its input voltages according to Eq. 2.3.

$$\Delta V_{fg} = \frac{Q_{fg} + C_D \cdot V_D + C_S \cdot V_S + C_B \cdot V_B + \sum_{i=1}^{n} \Delta V_i \cdot C_i}{C_{tot}} \tag{2.3}$$

where V_{fg} is the floating-gate voltage, $V_{D/S/B}$ and $C_{D/S/B}$ are respective terminal voltages and capacitances that are assumed to be constant [2]. Well-capacitance and well-bias can be considered to be one of the inputs. Now, the change in V_{fg} can be used to evaluate the respective change in drain current I_D according to the standard I_D versus V_{GS} equations.

Basically, an FGT can be either n-type or p-type. However, while both types are functionally similar, n-type transistors are not so well suited for some of the programming methods as explained in the next section. This is the reason why p-type FGTs are often preferred. For, example all FGTs presented in this thesis are p-type devices.

2.2 Programming Methods

The purpose of this section is to provide necessary information for the reader to facilitate the understanding of circuit topologies presented in the following chapters. Principles of different programming methods are discussed, and their usability on a general level is estimated. From this perspective, it is not necessary to go deeply into the physical mechanisms that these methods are based on, nor detailed mathematical models. A selected set of programming methods presented in this section is examined in more detail in Chapter 3, and verified through experimental measurements in Chapter 4.

2.2.1 Basics of Programming

The aim of the programming is to transfer charge carriers (electrons or holes) through $Si - SiO_2 - Si$ layer structure where the $Si - SiO_2$ interface introduces an approximately 3.15 eV energy barrier to electrons and 4.63 eV energy barrier to holes [19]. The effective width of the energy barrier is determined by the potential difference between the silicon layers. The carriers must gain sufficient energy so that they can either travel through or jump over the energy barrier (see Figs. 2.4 and 2.6). When the energy barrier heights for electrons and holes are examined, it can be readily seen, that less energy is needed to move electrons through the SiO_2 layer. Consequently, most methods are based on moving electrons through the insulator.

[2] This simplified equation neglects the fact that capacitance values and drain voltage may change according to the bias point.

The programming of FGTs can be dynamic or static. One example of dynamic programming is the Autozeroing Floating-Gate Amplifier (AFGA) [17], which adapts the floating-gate charge in response to the applied input signal. Static programming means that FGTs are programmed deterministically to a given target. An example of static programming is presented in [18], where the matrix coefficients needed in image processing are programmed on analog FGTs. There are certain applications that do not fall in either of these groups. For example, the amount of transferred charge itself can be the desired result, e.g. in dosimetric applications [16]. Another study [43] reports how the accumulated charge on FGTs can be used for fatigue monitoring in biomechanical implants. It was shown how the piezoelectric voltage generated by the sensor can be directly used as a source for the programming pulses thus making the monitoring sensor self-powered.

The exact amount of stored charge is relevant for analog applications, whereas binary programming (a lot of charge versus a little bit charge) suffices for digital applications. Thus, it is obvious that the transition from binary programming to continuous programming involves several issues that need to be addressed to achieve desired functionality with an acceptable level of accuracy. Flash-memories based on Multi Level Cells (MLC) are also available, for storing more than a single bit of information on an FGT. The programming of the MLCs is more complicated than the binary case, but needs far less accuracy than the pure analog, continuously valued memory. When programming to a target, the state of the FGT has to be measured: The programming is typically performed in cycles (the iterative process applies also to MLCs). One cycle includes both the programming phase and the measuring phase where the effect of the applied programming is evaluated. In principle any of the following quantities of the FGT, namely charge stored on the floating-gate Q_{fg}, floating-gate voltage V_{fg}, drain current I_D, drain voltage V_D or related quantity such as channel conductance can be chosen as the quantity to be monitored. However, in practice the easiest way is to measure the I_D.

Fig. 2.3 presents a circuit setup for an indirect FGT programming scheme [15] where the signal and programming transistors are separate devices. Another possibility is direct programming that uses the same transistor for signal and programming. Typically, more switches are needed in the direct programming scheme because the FGT must be disconnected from other circuitry during the programming phase and then reconnected for the run-time phase. These switches both increase the complexity of the circuit and introduce additional resistances and capacitances and thus decrease the performance. An indirect scheme permits in-circuit programming but increases the capacitance C_{tot} connected to the floating-gate. The direct consequence of the increased C_{tot} is that more charge has to be transferred for the same threshold voltage shift (see Eq. 2.1). It is noteworthy, that the

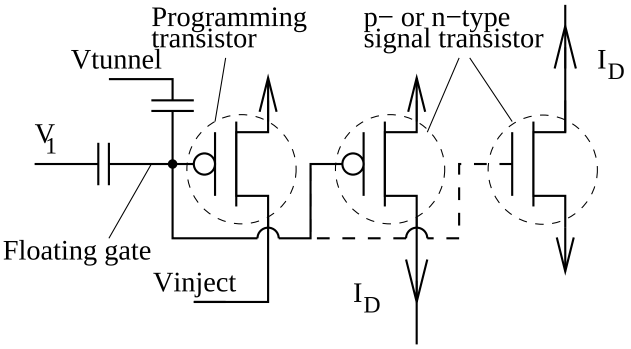


Figure 2.3: Schematic of an FGT with bidirectional programming capability. Fowler-Nordheim electron tunneling is used to remove electrons from the floating-gate and hot electron injection to add electrons to the floating-gate. Indirect programming scheme is applied with either p-type or n-type signal transistor.

indirect method allows transistors connected to a common floating-gate to be of different type. For example, the p-type signal transistor in Fig. 2.3 can be replaced with an n-type transistor.

Bidirectional programming is required in nearly all practical applications. The programming can be global or local, or a combination of these, i.e. global in one direction and local to the other. Furthermore, the programming mechanism can be the same or different in order to add charge (electrons or holes) to the floating-gate or to remove charge from it. On the other hand, some of the presented programming techniques can be used only to add electrons to the floating-gate and therefore need to be supplemented with some other method in order to achieve bidirectional programming.

In nearly all practical cases regarding multiple FGTs on the same chip, it is necessary to organize the programming scheme so that it allows a possibility to program each FGT independently, so that the programmed charge of neighboring FGTs remains unchanged. The difficulty in guaranteeing such selectivity with a given programming method depends on the number of parameters that need to be set for the programming.

It is important to understand that the transfer processes are quantum-mechanical by nature and therefore it is only possible to increase or decrease the probability of charge transfer rather than define strict limiting values for them. Essentially, it is the wave-like properties of charge carriers that enable them to cross the energy barrier. Understanding these effects on the atomic level would take us deep into quantum mechanics, and therefore, it is beyond the scope of this thesis. Instead, the principles of different programming methods on the macroscopic or qualitative level are briefly summarized.

2.2.2 UV Photo Injection

Exposing FGTs to UltraViolet (UV) light is perhaps the easiest way to neutralize the negative charge stored on the floating-gates. The mechanism is based on the ability of short wave ultraviolet light (high energy per photon) to impart enough energy (>3.15 eV) to the floating-gate electrons in the Si valence band to enter the SiO_2 conduction band. These excited electrons induce an electrical current through the oxide [20]. The UV Photo Injection method does not require high voltages nor extra capacitors connected to the floating-gate as many other methods, but needs an appropriate chip package to enable light exposure. The process is quite slow and depending on the light intensity needs several minutes or even hours (in ordinary sunlight) to erase the negative charge stored on the gate. It is not possible to add electrons to the floating-gate with the help of UV-radiation and hence it is a one-directional process. It should also be obvious that it is very difficult to design an FGT array that supports selective UV-light programming, because the mechanism is controlled by only two parameters: UV-light intensity and exposure time.

2.2.3 Fowler-Nordheim Tunneling

Fowler-Nordheim tunneling (F-N tunneling) [27], which is sometimes also called cold electron tunneling, is a process which allows electrons to penetrate through the energy barrier at the $Si{-}SiO_2$ interface. It is also possible to tunnel holes in a standard CMOS process [22], but only through an ultra-thin oxide layer (2-4 nm) because of the larger energy barrier height. This is in contrast to electrons, which are able to travel through much thicker dielectric. Such an ultra-thin oxide does not support long-term charge retention. Furthermore, only standard CMOS processes having a $Si - SiO_2 - Si$ at the gate are considered within this thesis. Hence, hole tunneling is beyond the scope of this thesis. The fundamental reason why the FGT becomes leaky if the dielectric is made too thin is that, an ultra-thin dielectric allows so called direct tunneling [23]. In this process electrons can spontaneously travel through the gate oxide almost independently of the electrical field, see Fig. 2.4.

A higher energy level than 3.15 eV is required for electrons to pass over the energy barrier Φ_B[3], however the barrier width is also relevant. At room temperature, the thermal energy level of electrons is sufficient for penetrating approximately 5 nm into the oxide. In the absence of the electric field, the electrons will fall back to the silicon (or polysilicon) if the oxide layer is wider than 5 nm. However, the effective barrier width can be reduced

[3] Interfacial barrier heights may be unequal for both sides of the dielectric. However, for rough calculations it is reasonable to assume that $\Phi_{B1} = \Phi_{B2}$, see Fig. 2.4.

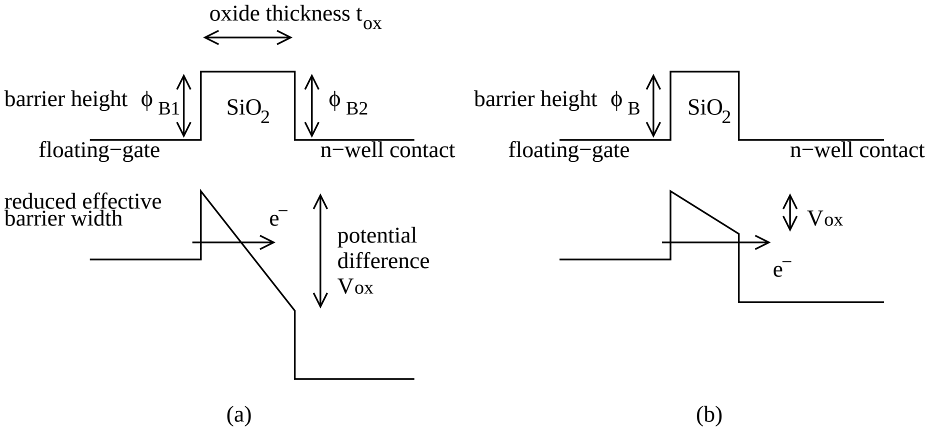


Figure 2.4: A conceptual energy band diagram of the physical differences between F-N and direct tunneling. (a) If the SiO_2 layer is thick enough, electrons have almost negligibly small probability to escape the conduction band of the floating-gate. When V_{ox} is elevated from zero (the upper figure) to a sufficient level ($V_{ox} > \Phi_B$) (the lower figure) the effective barrier width is reduced so that electrons can tunnel through the triangular-shaped energy barrier. (b) In direct tunneling the electrons can tunnel through the trapezoidal-shaped energy barrier even if the $V_{ox} < \Phi_B$.

by applying an electric field across the oxide. Thus, in this process the energy level of electrons is not elevated but the effective barrier width is reduced allowing electrons to tunnel through the energy barrier. This mechanism enables electron tunneling at a relatively low current density. Hence, the name cold electron tunneling, albeit the electrons do become hot (their energy increases) when entering the oxide.

The potential difference V_{ox} across the oxide decreases the effective barrier width so that electrons can pass through the triangular-shaped barrier into the oxide conduction band, see Fig. 2.4. From the oxide conduction band, these electrons are swept over to the side of the oxide determined by the polarity of the electric field. Naturally, a positive voltage relative to the floating-gate voltage attracts electrons and negative voltage repels them. To initiate significant current flow through the oxide, an electric field larger than 6.4 x 10^8 V/m must be applied across the oxide layer [29]. This information can be used to estimate necessary V_{ox} for a certain oxide thickness.

F-N tunneling is very sensitive to gate oxide thickness t_{ox} because of its exponential relationship to current flowing through the dielectric. Several empirical or semi-empirical equations presented in the literature describe the current flow through the SiO_2 under F-N tunneling conditions. One example is given below [28].

$$J = K_1 F^2 \exp(\frac{-K_2}{F}) \tag{2.4}$$

where J is the current density, F is the electric field across the SiO_2 and K_1 and K_2 are constants determined from the measured data. Typically, F can be approximated by V_{ox}/t_{ox}. K_1 and K_2 are in the order of 1 μA/V^2 and 200 MV/m for oxide thickness > 5 nm. The distinct current density may be higher or lower than the one predicted by Eq. 2.4, because of the trapped charge in the insulator or the geometrical shape of the tunneling junction, which contributes to the tunneling efficiency [29].

Hence, electron tunneling requires only two parameters (voltage across the oxide and the tunneling time) to be set. On the other hand, using F-N tunneling for selective programming of an FGT array can be expensive in terms of area and power due to the required high voltages. The area of the high-voltage switching circuitry can become large and complex in order to properly isolate the programmed and non-programmed devices. This is one reason why F-N tunneling is typically used for global erase.

It is not practical to tunnel electrons through the signal transistor's oxide, because adding high-voltage switches to the signal path increases the complexity and area, deteriorates the signal to noise ratio and can cause undesirable modifications to the functionality of the device. Therefore, a tunneling junction should be added to the floating-gate. The capacitance of the tunneling junction must be small enough relative to C_{tot} in order to achieve a sufficient electric field with a reasonable tunneling voltage.

Electrons can be added to and removed from the floating-gate by the tunneling mechanism because the direction of electrons is exclusively determined by the polarity of the electric field across the oxide. Hence, in principle one tunneling junction suffices for bidirectional programming. However, this approach may lead to difficult challenges in practical implementations. This topic is addressed in more detail in Chapter 3. Alternatively, two capacitors of sufficient capacitance ratio can be used to realize a bidirectional tunneling scheme. The operation principle of such a programming scheme is shown in Fig. 2.5, where $C2$ is a control capacitor that sets the floating-gate voltage V_{fg} while $C1$ provides a tunneling junction. $C2$ dominates the $Ctot$ so that V_{fg} is a strong function of $V2$ and a weak function of $V1$. Thus, according to Eq. 2.3 $V_{fg} \approx V2$. Consequently, the polarity of the electric field across $C1$ can be reversed by reversing the voltages applied to $C1$ and $C2$. In addition, it must be taken into account that the parasitic capacitance Cp decreases the effective tunneling voltage.

Using two capacitors offers an extra advantage of better support for selective programming, because adding an extra capacitor effectively equals adding an extra parameter. The weakness of this technique is that the gate

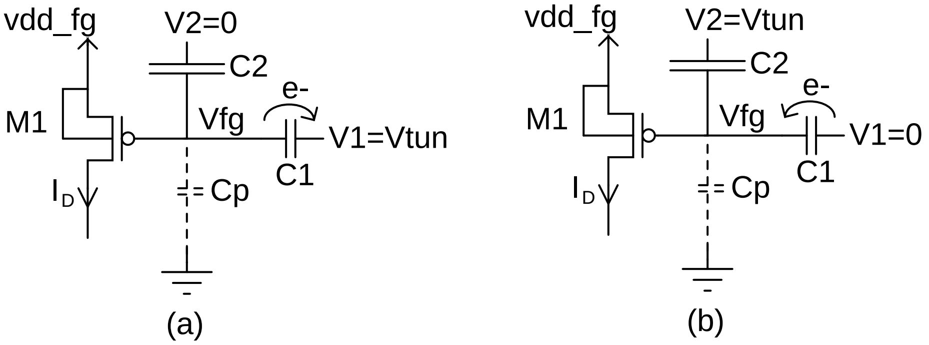


Figure 2.5: The principle of bidirectional tunneling scheme. (a) electrons are removed from the floating-gate due to $V1 >> V_{fg}$. (b) electrons are added to the floating-gate due to $V1 << V_{fg}$.

oxide of the signal transistor $M1$ is exposed to a high electric field when V_{tun} is applied to $C2$, see Fig 2.5 (b). The tunneling current through the gate of $M1$ can be minimized by disconnecting the drain of $M1$ of its run-time circuit and coupling it to vdd_fg for the programming phase. Furthermore, if $M1$ is replaced with an n-type transistor, also source (and bulk) should be connected to vdd_fg for the programming phase in order to minimize the electrical stress of $M1$.

2.2.4 Hot Electron / Hole Injection

When electrons or holes are accelerated in an electric field their energy increases. Energetic carriers are often called hot because thermal energy E and temperature T are related through Boltzmann's constant k: $E = kT$. These energetic carriers can be injected into the oxide to become trapped charge, drift through the dielectric layer (SiO_2), create interface trapped-charge, or generate photons [33]. Many of these events are unwanted in normal operation because they cause undesirable device behavior modifications such as threshold voltage shift and mobility degradation. However, charge transfer by hot electron / hole injection through the SiO_2 layer can be exploited in the programming of FGTs for intentional threshold voltage modification.

There are several mechanisms that can give rise to oxide current when a MOSFET is biased under hot-carrier conditions [23, 34]. These are Substrate Hot-Electron (SHE), Auger Recombination (AR), Secondarily Generated Hot-Electron (SGHE), Channel Hot-Electron (CHE) and Drain-Avalanche Hot-carrier (DAHC). Occurrence of significant hot-carrier effects in silicon requires electric fields greater than 4 x 10^6 V/m [23]. This is roughly 150 times lower than that required by F-N tunneling, because the kinetic energy of the carriers is exploited in the injection process. Hot carriers have enough energy to pass over the energy barrier, see Fig. 2.6. Unlike electron tunneling, hot electron injection is a one-way process (carriers can be accelerated only on the channel side). That is, it can only be used to add electrons to the floating-gate. While in principle, it is possible to inject hot holes to the floating-gate [35], in practice it is very hard because of the higher barrier height for holes and because hot holes tend to generate hot electrons by impact ionization [4], which can more easily pass over the barrier. In the following, different hot carrier generation mechanisms are discussed briefly.

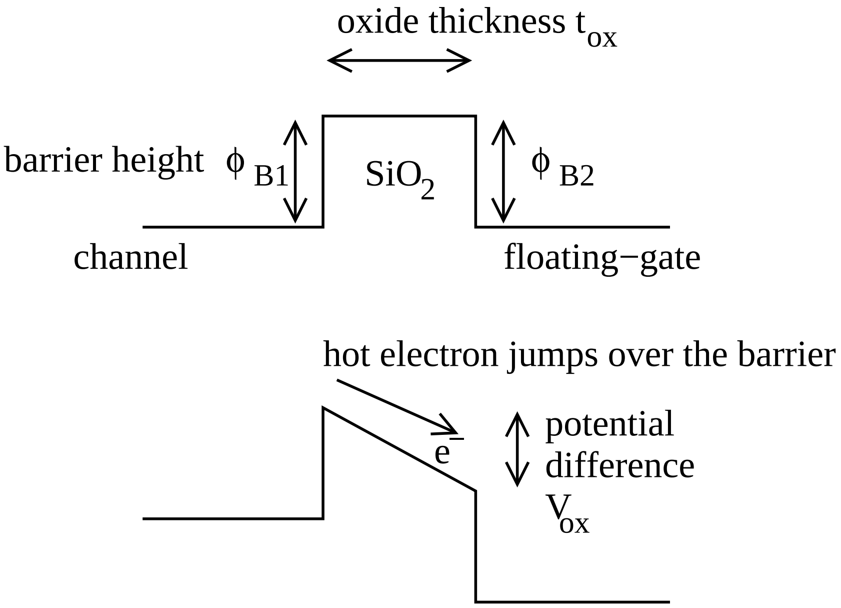


Figure 2.6: A conceptual energy band diagram of the hot electron injection. The kinetic energy of hot electrons exceeds the barrier height Φ_B so that electrons can pass over the barrier. Required vertical field V_{ox} is within the normal supply voltages, but sufficiently high (typically lateral) field is required to generate hot electrons. It is possible to accelerate i.e. generate hot electrons on the channel side only. Consequently hot electron injection is a one-way process and can be used only to add electrons to the floating-gate.

SHE is mainly used for evaluation of the gate oxide quality. It resembles F-N tunneling, but hot carriers must be externally generated (for example thermally) or injected from a forward-biased pn-junction. The need for thermal excitation or varying substrate bias (back-bias) to control the energy of hot electrons explains why this mechanism is not practical for programming an array of FGTs. Above a certain value of the floating-gate voltage, F-N

[4] The result of the impact ionization is two holes and one electron.

tunneling starts to dominate the gate current and the contribution of SHE is masked.

In the Auger process, two carriers recombine and give their energy to a third carrier. In principle, this mechanism allows gate current with drain voltages below the barrier height. However, to reach a sufficient energy level to surmount the barrier, the recombination energy of carriers in equilibrium with the lattice does not suffice [34]. Hence, in practice either the recombining carriers or the receiving carrier must be hot. However, the probability of recombination decreases exponentially with the energy of the carrier. Therefore, it is very difficult to take advantage of this effect to inject carriers to the floating-gate. However, it is probable that the AR enhances the CHE injection for large drain voltages.

SGHE is the most complicated among the HE-processes, involving many overlapping mechanisms for secondary hot carrier generation, such as secondary induced electrons by the hot hole substrate current, photon induced injection and bipolar injection. Basically, the bias conditions in SGHE are similar to those in CHE and in DACH (discussed in the following paragraphs) except for the influence of the substrate's back-bias. The need for back-bias along with the control of many overlapping injection mechanisms simultaneously makes SGHE less attractive for the programming of FGTs.

In the case of CHE, the accelerating electric field is lateral. Hot electrons must be collected into the floating-gate by a properly oriented vertical field. In an n-channel transistor, the combination of low gate voltage V_G and high drain to source voltage V_{DS} (high lateral field) creates a large number of hot electrons near the drain. However, high gate voltage ($V_G > V_D$) is necessary for a high vertical field that collects part of the hot electrons to the gate. Therefore, it is not possible to maximize both the lateral and the vertical field simultaneously. From this contradiction it follows that for an n-channel MOSFET the injection efficiency peaks when V_G is approximately equal to V_D assuming that V_S and V_{SUB} are grounded. However, most of the effects caused by CHE are considered harmful in "normal" use of transistors. As these effects are pronounced as the scaling continues, there are several process-control techniques, such as Lightly Doped Drain (LDD), double diffused MOSFET structure and replacing the SiO_2 with the Si_3N_4, that specifically work to avoid CHE [14]. Consequently, the precise programming of nMOS transistors by using CHE is virtually impossible [15].

If the accelerating lateral field is high enough and ($V_G < V_D$), hot electrons and holes are generated by impact ionization and avalanche multiplication at the drain (DAHC). The gate current of the DACH as a function of V_G is not monotonic because the injection efficiencies of hot holes and hot electrons peak at different points, as can be seen from Fig. 2.7. However, the peaks are not equal in height. For an n-channel device, the gate current caused by the injected holes is lower due to the higher energy barrier to the

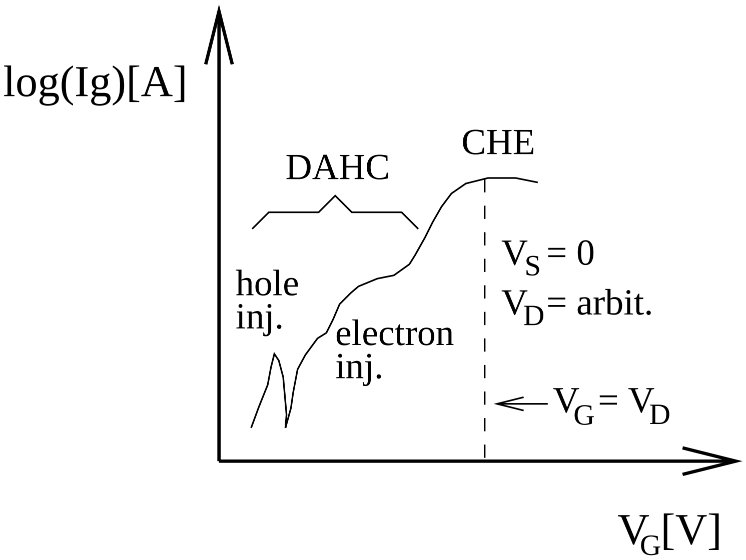


Figure 2.7: An illustration of I_G against V_G for an n-channel device in DAHC for arbitrary V_D (adapted from [34]). There is a smooth transition from the DAHC to the CHE when the V_G becomes comparable to the V_D.

holes in the $Si - SiO_2$ interface. Although the vertical field of an n-channel device does not favor the collection of electrons to the gate, a part of the ionized hot electrons can be injected against the repulsive field. On this account, the efficiency of DAHC injection compared with CHE injection is lower. When the V_G is elevated further, the impact ionization decreases and the number of hot electrons in the channel increase, so that there is a smooth transition from the DACH to the CHE.

In a p-channel device, the hot electrons generated by the DAHC mechanism can be injected to the gate. This mechanism is often referred to as Impact Ionized Hot Electron Injection (IIHEI) [36]. The advantage of the IIHEI mechanism is the properly oriented high vertical field under DAHC bias conditions as opposed to nMOS devices under CHE or DAHC bias conditions. This makes IIHEI amenable for the precise programming of a pMOS FGT. Furthermore, the achieved gate current is larger for the p-channel device than for the n-channel device under comparable bias conditions (DACH) due to the smaller energy barrier to the electrons.

The modeling of IIHEI requires the extraction of empirical parameters. For example, a semi-empirical equation of IIHEI gate current was given in [10]

$$I_g = \alpha I_s \exp\left(\frac{-\beta}{(V_{gd}+\delta)^2} + \lambda V_{sd}\right) \tag{2.5}$$

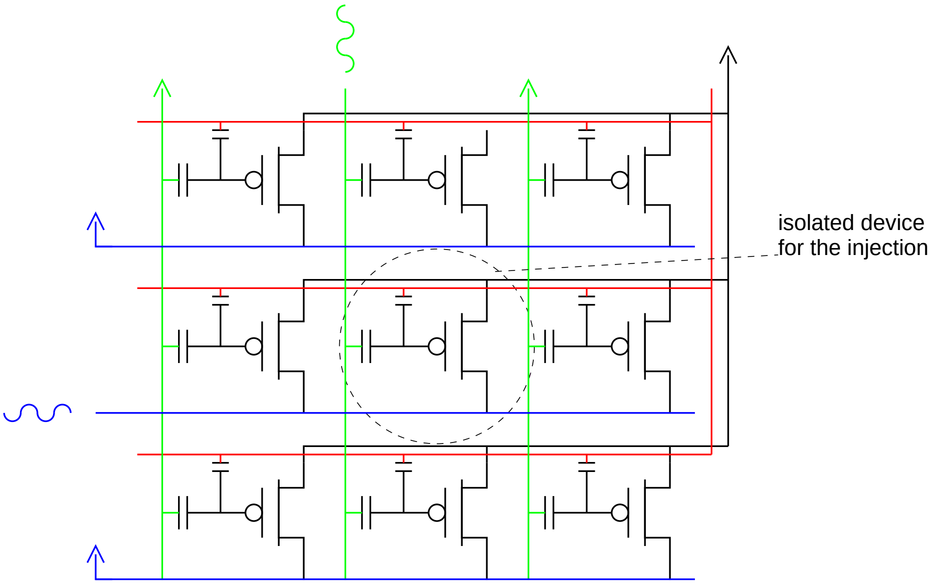


Figure 2.8: The programmed FGT can easily be isolated from non-programmed devices during injection.

where I_g is the gate current, I_s is the source current, V_{gd} is the gate-to-drain voltage and V_{sd} is the source-to-drain voltage. α, β, δ and λ are parameters determined from the measured data. From this equation, the dependence of the gate current on the vertical (V_{gd}) and the lateral (V_{sd}) electrical fields is evident. A more rigorous study of IIHEI with injection efficiency considerations is presented in [37].

In practice, the most useful injection mechanisms are CHE and DACH (or IIHEI), because there is no need for the back-bias. Utilizing these carrier injection mechanisms requires the control of two electric fields simultaneously. With CHE the coexistence of lateral and vertical fields typically makes the I_g vs. V_g curve parabolic, because when one field is strengthened the orthogonal field gets weaker. Optimal injection efficiency is obtained by balancing the hot carrier generation efficiency with the vertical field that transports the hot electrons to the floating-gate.

Voltage controlled carrier injection needs three parameters to be set: time, V_{DS} and V_{GD}. V_{DS} and V_{GD} can be regarded as equivalents to word line and bit line in standard memory architecture. This is advantageous for the selective programming of an array, since the programmed device can be easily isolated, see Fig. 2.8. However, considering especially an analog FGT array, the three parameters need to be controlled simultaneously while taking into account the initial charge. This means that accurate control of electron injection mechanisms is challenging in practical circuits.

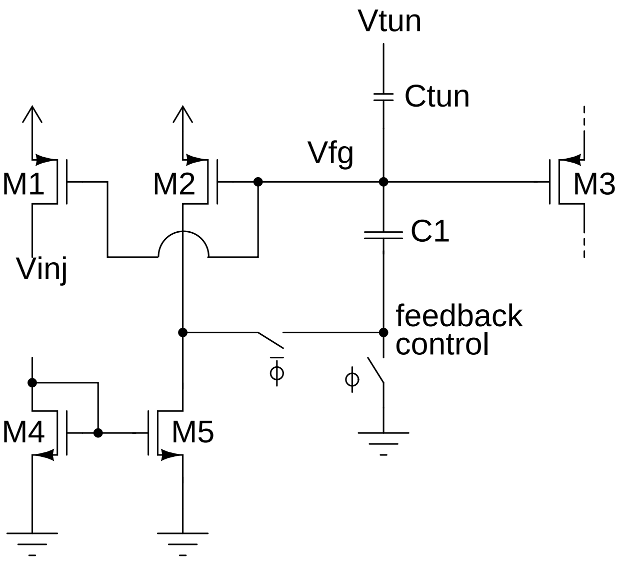


Figure 2.9: A simplified schematic of a current biased injection scheme.

There is also an alternative electron injection programming scheme, which does not require accurate control over V_{GD}. According to Eq. 2.5, the gate current I_g is proportional to the source current I_s. Typically, I_s is controlled with gate voltage bias V_{GD}, but current bias can be used as well. A simplified schematic of such an alternative programming scheme [38, 39] is shown in Fig. 2.9. During normal operation, $Vtun$ and $Vinj$ are set to *gnd* and Vdd, respectively. Transistors $M4$ and $M5$ form a current mirror that biases $M2$ with V_{fg}. During programming, a feedback loop is established between the drain and gate of $M2$. As $C1$ dominates C_{tot}, the current set by $M5$ flows through $M2$, independently of Q_{fg}. This results in a constant current through $M1$ as it will mirror the current of $M2$. The change in charge Q_{fg} will be a function of the bias current of $M1$, the $Vinj$ applied to $M1$, and the duration of the injection pulse.

The current biased injection scheme needs less control in the programming mode than the gate voltage controlled scheme, because a sufficient current bias is determined in the design phase. However, simplified control as compared to the gate voltage controlled scheme is traded for additional power consumption that depends on the chosen current bias, additional switches (increased area) for controlling the feedback loop and loss of one degree of freedom for the gate current adjustment. Also, for these reasons, it is not that amenable for programming large FGT arrays as is the gate voltage controlled scheme. A current bias controlled injection scheme was experimentally tested and the obtained measurement results are discussed in Chapter 4.

While CHE suffices for digital programming, IIHEI is better for accurate hot electron injection for analog applications. This conclusion is based on two things: 1) unintentional CHE is considered harmful and consequently its utilization for programming purposes is efficiently prevented by device level manufacturing optimization 2) the vertical field is incorrectly oriented for supporting CHE but properly oriented for IIHEI. Furthermore, whereas CHE is efficiently prevented, the IIHEI will be available in all DSM CMOS processes because it can not be eliminated without affecting basic transistor operation. Therefore, all FGTs in this thesis are p-channel devices employing the IIHEI mechanism for electron injection. Certainly, the better control of p-channel device programming is traded for lower performance in terms of current gain, transconductance, and speed, in comparison to n-channel devices. However, it is possible to circumvent this trade-off by using an indirect programming scheme, in which a pMOS is used for programming and nMOS acts as a signal transistor.

Finally, some aspects related to injection mechanisms on a system level are considered. While a similar mechanism to the CHE, Channel Hot-Holes (CHH) injection, does exist for p-channel devices, it is not useful due to very low injection efficiency. Thus, in practice, electrons can not be removed from the floating-gate with injection mechanisms. Consequently, any bidirectional programming scheme utilizing a hot electron injection mechanism needs a complementary F-N electron tunneling mechanism to remove electrons. It should also be noted that an injection mechanism, which needs a high lateral field to generate hot carriers, may also limit the use of some important circuit techniques, for example cascading.

Typically, an FGT is "ramped up" before applying injection pulses. That is, all of the voltages capacitively coupled to the floating node (drain, source, control gate, tunneling junction) are increased in small steps to avoid large potential differences between any nodes. After injection, all terminals are ramped down respectively to observe the change caused by the threshold voltage shift. Ramp up is required when only positive voltages are available to get an adequate drain source voltage for injection. If negative voltage pulses are applied to the drain terminal during injection there is no need for ramp up nor ramp down.

An important long-term effect due to repeated programming is that the oxide wears out. This happens regardless of the selected programming mechanism. In practice the wear out is due to trapped charge and it causes increased gate currents for the same programming parameters and initial state. Another phenomenon that affects the FGT in the long-term causing charge loss is thermionic emission [40]. When the temperature is increased, the energy of charge carriers (sometimes referred to as "thermions" in the literature) is increased so that some of them are able to overcome the energy barrier of the $Si - SiO_2$ interface. In addition to long-term effects, a short-term drift can be observed every time FGT is programmed. The short-term drift is due to trapped charge in the oxide and at the interfaces settling to a new equilibrium state after programming. Both of these effects should be taken into account especially when designing and evaluating systems which include analog FGTs.

2.3 Performance Aspects of FGTs

More often than not, analog IC design is based on matching devices. The usual way to get a sufficient level of matching is simply to spend more area, thereby increasing the capacitance and power consumption (assuming a fixed operating speed). Respectively, the operating speed (or bandwidth) can be boosted by using wider transistors which also leads to increased power consumption. With FGT technology it is possible to partly decouple the bandwidth-accuracy-power trade-off, since the programming feature of FGTs enables accuracy or matching improvement without the usual proportional increase in the device size and power consumption. The programmability of FGTs also helps in removing extra overhead needed in traditional analog circuit design for linearity improvements, offset removal etc. Naturally, the area needed for the programming infrastructure must be considered when evaluating the trade-offs.

The performance of an FGT is influenced by the unavoidable capacitive divider at the input. The major drawback of the capacitive divider at the input is that it inevitably leads to degraded gain-bandwidth products. This is because the input capacitors act as coupling capacitors which degrade the signal path as the frequency increases. That is why FGTs are in general more suitable for systems with low frequency and bandwidth. On the other hand, [50] reports an FGT-based frequency divider that reaches the -3 dB cut-off frequency at approximately 3 GHz. Even though only simulations were presented without measured results, it can be seen that high frequency applications are feasible. The discussion of the frequency response of the AFGA (Section 4.3) brings additional insight into this issue.

Besides operation speed, there are other performance-related questions of fundamental importance that need to be answered when the use of floating-gate devices is considered. For example, what happens to the Signal to Noise Ratio (SNR) when the gate is isolated by an extra capacitor? This topic is addressed in [51] and those results are briefly summarized in the following. Additional considerations from the technology scaling point of view are addressed in Chapter 3.

Small signal models of a regular pMOS and a p-type FGT are used as the starting point of this analysis. For simplicity, stand alone devices are assumed so that both devices drive similar loads. The transconductance

g_m of a pMOS transistor in saturation region (and above threshold) can be derived from the small signal model of a transistor (see Fig. 2.10) and is given as

$$g_m = k'_p \frac{W}{L}(V_{GS} - V_{th}) \tag{2.6}$$

where k'_p is a process transconductance parameter given by

$$k'_p = \mu_p C_{ox} \tag{2.7}$$

where μ_p denotes the mobility of holes in the induced channel. An extra capacitor $C1$ at the input of the FGT device acts as a decoupling capacitor and forms a capacitive divider between the input capacitor and all other capacitors connected to the floating-gate. The ratio of the capacitive divider is given as

$$\kappa = \frac{C1}{C_{tot}} \tag{2.8}$$

so that the transconductance of an FGT $g_{m,FGT}$ is given as

$$g_{m,FGT} = \kappa g_m \tag{2.9}$$

As opposed to a regular MOS transistor, an FGT device offers an extra degree of freedom for g_m adjustment by manipulating the effective threshold voltage that affects the overdrive voltage.

The intrinsic cutoff frequency f_T is defined as the value of the frequency at which the short-circuit current gain of the device drops to unity (i.e $Iout/Iin = 1$, see Fig. 2.10). The same degradation factor, κ, applies to the cutoff frequency, because f_T is directly proportional to g_m. Thus, for an FGT device $f_{T,FGT}$ is given as

$$f_{T,FGT} = \frac{\kappa g_m}{2\pi(C_{gs} + C_{gd} + C_{gb})} \tag{2.10}$$

where C_{gs}, C_{gd} and C_{gb} are intrinsic capacitances of a MOS device (see Fig. 2.1).

In this simplified analysis only the changes due to the input capacitor $C1$ were considered. However, it is straightforward to extend the analysis to FGTs with multiple inputs. For example, the tunneling capacitor appears in parallel with the C_{gs} in the small signal model, and thus adds one linear term to the denominator of Eq. 2.10, further decreasing the $f_{T,FGT}$ as can be expected.

Similar definitions can be derived for a transistor operating in subthreshold or in a linear region yielding the same dependence on κ. The obvious consequence of $C1$ is that the input signal amplitude is reduced by a factor κ, so the input signal power is $\frac{1}{\kappa^2}$ smaller than that of a nominal MOS. On

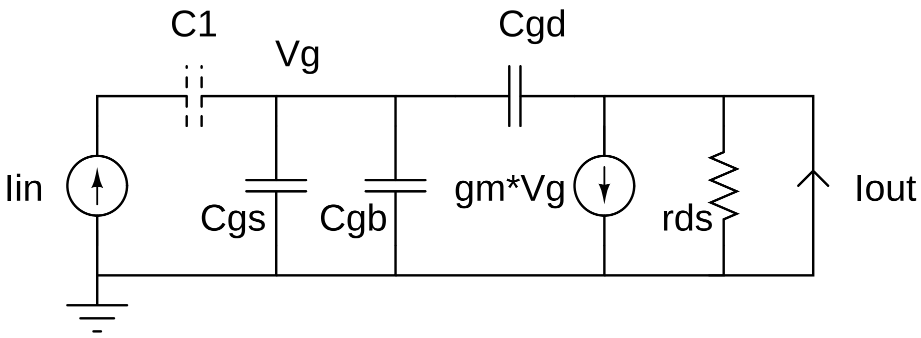


Figure 2.10: Small signal model (pmos) for defining the intrinsic cutoff frequency f_T. Here, it is assumed that the source and the substrate are at the same potential. A nominal MOS device does not include capacitor $C1$ which exists only for the FGT device.

the other hand, the noise power is also reduced by the same factor. The result is that for comparable device sizes the SNR of an FGT is similar to the SNR of a nominal MOS device. Furthermore, the flexibility advantage of the FGT over the regular MOS device can often compensate for the small losses in performance. Here, only the pMOS device was considered, but with indirect programming methods it is possible to combine the flexibility of an FGT device with the better relative performance of an nMOS over a pMOS device.

2.4 FGTs and Circuit Simulators

A couple of notes are made about simulating the FGTs with different circuit simulators such as SPICE, SPECTRE, ELDO. In principle, simulating circuits containing floating-gate transistors is quite simple. For example, to set up a transient simulation, a large resistor for example $R = 1e23$, must be added between the floating gate and ground. Additionally the initial condition (V_{fg0}) for the floating node must be set. The resistor is needed because the simulator requires a DC path to ground to find a converged solution for a DC analysis. The set initial condition is for modeling the programmed charge.

However, typically the circuit models provided are as such unable to model the gate currents under programming conditions correctly when the "thick" oxide devices are considered. This is understandable, because the gate currents are negligible when the transistors are operated within normal supply voltages. To model the gate currents due to F-N tunneling and IIHEI, voltage dependent current sources need to be used in parallel with the transistors. However, the extraction of some technology dependent parameters is required to implement a realistic simulation model that matches with the empirical measurement results [10]. For “thin” oxide devices the models are more accurate simply because non-negligible gate currents within the nominal supply voltages must be taken into account.

Overall, lack of accurate simulation models makes high performance FGT circuit design challenging, but the programmability feature can often alleviate this shortcoming.

Chapter 3

Migrating FGTs to Deep Sub-Micron Standard CMOS processes

The growth in sales of mobile and wireless electronic devices is a long trend that runs on. The growth comes both from existing desktop devices being converted into mobile versions and emerging completely new devices. Technology scaling along with other development steps has reduced the power consumption of devices significantly. However, technology scaling poses several challenges for the implementation of non-leaky floating-gate devices. Nevertheless, a successful integration of analog FGTs on a DSM CMOS can be highly beneficial, especially for some new power critical applications, justifying this exercise. One case study will be briefly discussed before addressing the compatibility issues.

3.1 Low-Power Subthreshold Electronics

For mobile devices the power consumption versus performance and form factor are critical parameters. The importance of power consumption will get further pronounced as the next generation wireless sensor networks appear. Such networks often target energy autonomous operation, an example being the Body Area Network (BAN).

The purpose of a BAN is to monitor the vital signs of the body [41]. The BAN can make alerts of acute changes in system activity and sense hidden symptoms of incipient serious diseases, such as hidden arrhythmia, which can be revealed only through long-term monitoring. Minimizing the energy consumption is crucial for systems like BAN, which try to harvest all the required energy from their surroundings. Things like data representation and processing affect the amount of transmitted data. Therefore, the whole sig-

Technology	$180ULL$	$90LP$	$40G$
Dynamic power consumption	66 $\mu W/MHz$	12.5 $\mu W/MHz$	4 $\mu W/MHz$
Floorplanned area	0.11 mm^2	0.03 mm^2	>0.01 mm^2

Table 3.1: ARM Cortex-M0 Implementation data [52].

nal processing chain must be reconsidered carefully when the power budget is extremely tight. On the other hand, despite the limited power budget, the system reliability should be very high. This is a big challenge that has inevitably tremendous consequences on the circuit design. By adopting more efficient power management schemes and low power circuit topologies, battery lifetime can be extended. In general, the best alternative would be if the system could automatically adjust the performance according to the available power resources.

For energy-autonomous applications, in addition to power consumed when doing something useful, special attention must also be paid to minimizing the leakage currents when the system is idle. Nowadays, this static power consumption constitutes an increasingly significant part of the total power consumption. IC foundries have risen to the challenge by including devices devoted for low-power digital design in the design kit. Hence, digital designers typically have a wider range of devices to choose from to match the needs of a given task. For example, a low-power version of basic transistor may have a somewhat thicker gate oxide and a different doping profile than the general purpose version of the same device. Thus, it has a smaller leakage current under the same bias conditions.

Digital hardware benefits directly in terms of area and power consumption from the migration to smaller line width processes. The power consumption and silicon area of the popular ARM processor are presented in table 3.1 for three different technology nodes. It shows that halving the line width of the implementation technology results in cutting the area and the dynamic power consumption into roughly one fourth of the original. It should be noted that the scaling involves many other aspects, such as increased temperature (assuming fixed clock frequency) as the transistors are packed into ever smaller area. The principles of low-power digital design fall outside the scope of this thesis, however. On that subject, an interested reader should consult textbooks such as [42].

The power consumption and throughput of digital hardware scale linearly with the clock frequency. Therefore, preprocessing of the data in the analog domain can be highly beneficial, if it permits a considerably lower clock frequency. Herein the use of FGT technology can be particularly beneficial. The use of FGTs can potentially improve the reconfigurability and the power efficiency of the analog system. An FGT-based system could challenge the conventional approach (high precision ADC and DSP) in applications such as BAN where extreme power efficiency is required.

A comprehensive study of FGT-based power efficient circuits is presented in [51]. It presents techniques for using FGTs to implement power efficient signal processing systems in the analog domain. In this study, the signals are processed as much as possible in the analog domain before converting them into digital. That way expensive wideband ADCs and DACs can be replaced with cheaper baseband data converters and power intensive high speed DSPs with baseband DSPs. The promise is that, while the power consumption of the ADC, DAC and DSP is reduced, the corresponding increase in the power consumption of analog hardware is smaller. Therefore, the coexistence of analog and digital (hybrid computation) maximizes the computational ability, while prolonging the battery life.

The need for energy efficient electrical devices has stimulated great interest toward subthreshold signal processing, which is one of the key techniques for reducing power dissipation in both analog and digital integrated circuits. In this region, the energy per operation can be reduced up to an order of magnitude. Subthreshold operation is particularly suitable for low frequency applications such as BAN. This is because the bandwidth available per ampere of current consumed in the transistor is maximal in the subthreshold region. However, operating essentially with leakage currents means that circuits are more susceptible to noise and mismatch, making the design more challenging.

While lowering energy consumption is the primary motivation for subthreshold operation, it also has other useful characteristics. These include a high dynamic range that spans over six orders of magnitude and higher relative gain compared to above threshold operation. Both of these characteristics originate from the exponential I_D vs. V_{GS} dependency in the subthreshold region instead of the square law dependency observed above threshold.

Further evidence supporting the hypothesis that a hybrid computation would be optimal for ultra-low-power applications is found by examining the area and power costs as a function of SNR. The implementation cost curves illustrated in Fig. 3.1 (adapted from [42]) suggest that the implementation efficiency of analog systems will be higher than that of digital systems for low to moderate SNR levels. This is based on the facts that: 1) a single device in the analog computation system offers a much richer set of basis functions and 2) feedback loops used in analog systems greatly improve the SNR and robustness for only a slight increase in area and power. The power and area of an analog system increases steeper than a digital system. This is due to the lack of signal restoration in analog systems, since unlike in digital computation; it is in general not known where to restore the signal to in analog computation.

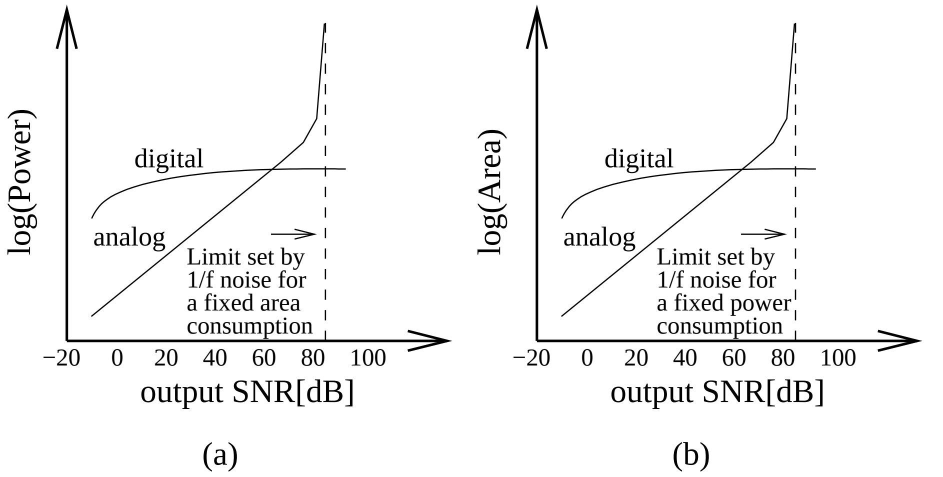


Figure 3.1: The implementation cost curves for the analog and the digital approaches, adapted from [42]. (a) illustrates the power cost and (b) the area cost.

To summarize, a successful implementation of a complete system for an ultra-low power application, such as a BAN compliant sensor node, requires:

- A mixed signal SoC, because a purely digital approach based on high speed, high resolution ADCs and DACs consumes too much power. On the other hand, functionalities such as power management and complex signal processing can more practically be implemented with digital hardware.

- DSM CMOS process, due to significant reduction in dynamic power consumption of digital hardware.

- Applying low-power techniques (subthreshold) throughout the design.

Certainly, achieving this aim would be easier if DSM compliant FGTs were available, since the properties of the FGT match very well with the challenges of analog signal processing in the subthreshold region. For example, programmability enables canceling the mismatches, and allows the tuning of the transconductance with high precision irrespective of the dimensions of the transistor. Above all, the FGT is practically the only device capable of implementing a non-volatile memory (analog or digital) in CMOS. On the other hand, in extremely low-power systems special attention must be paid on programming the FGTs, since the power efficiency advantage of FGTs can be lost by the programming infrastructure. That is to say, the generation of high voltage programming pulses consumes a lot of energy, especially if the programming needs to be frequently active.

3.2 Implementing FGTs in DSM CMOS — Getting Started

The practical implementation of FGTs in a DSM CMOS involves several challenges that need to be addressed. The first issue is that the availability of suitable devices for the implementation can not be taken for granted. Furthermore, it can be difficult to create a robust programming scheme that satisfies the design rules imposed by the foundry even though there are suitable devices available. Finally, careful design choices and a deep understanding of the physical structure are required in order to avoid unintentional parasitic effects and to keep the designed devices alive on the whole. While in principle all these are true regardless whether a DSM CMOS process is used or not, it becomes increasingly difficult when migrating FGTs to DSM process due to several reasons, which are addressed in the following.

Most modern general purpose standard (digital) CMOS processes offer transistors with multiple gate oxide thicknesses. The thick oxide devices are typically used in the IO-cells to make the IC compatible with the higher system voltage than that of used in the core design. However, these IO-transistors can be used for the core design as well, especially in the analog parts of the design. This is a good starting point to design FGTs featuring as long retention time as possible in a scaled down process. Unfortunately, all of the offered oxide thicknesses may not be available simultaneously on the same chip due to processing limitations. The first challenge arises from this limitation.

Technology scaling implies that the bulk doping increases as the channel length decreases [49]. Otherwise, the combined width of the drain and source depletion regions would be greater than the minimum channel length, which is an unacceptable situation leading to punch-through. The thin and thick oxide devices have different doping profiles and consequently the minimum channel length (L) and width (W) for devices with thicker oxide are somewhat larger than for those using minimum oxide thickness allowed by the process kit. An important consequence from the FGT technology point of view is that the junction breakdown voltage is lowered when the doping is increased. However, simultaneous scaling of gate oxide thickness means that the junction breakdown voltage remains greater than the voltage needed for the F-N tunneling. However, the margin between effective F-N tunneling and junction breakdown voltages is decreased.

How does the technology scaling affect the $g_{m,FGT}$, $f_{T,FGT}$ and SNR discussed in Section 2.3? As CMOS technology is scaled down, the gate-oxide capacitance C_{ox} per area increases as t_{ox} decreases. Let us assume that only the t_{ox} is scaled down and the device size remains fixed. A key observation is that κ remains constant because the total capacitance C_{tot}

scales with the same ratio as the input capacitance $C1$, so the performance relative to nominal MOS device remains unchanged. From increase in C_{ox} it follows that also k'_p increases because they are directly proportional by definition and so does the g_m. However, C_{gs}, C_{gd} and C_{gb} are also directly proportional to C_{ox}, so the transistor's intrinsic speed f_T hardly changes over technology [11]. This leads to a conclusion that due to technology scaling, the performance of an FGT is not severely degraded when compared with the nominal MOS device with a similar size.

3.3 Feasibility of Programming Methods in DSM CMOS

UV erasure is not feasible for advanced sub-micron technologies with multiple metal layers and copper backend, because of the limited UV transparency of the dielectric stack [21]. On this account, this method was not experimentally tested within this study.

There is no technological limitation that prevents programming of FGTs with F-N tunneling in the DSM CMOS processes. However, the thickness of the "thick" oxide in DSM technologies has come to the limit in which the probability of direct tunneling through the SiO_2 (app. 5 nm) is no longer insignificant. That is, the electrons have enough kinetic energy at room temperature to surmount the energy barrier regardless of the intensity of the vertical field (see Fig. 3.2). The gate leakage is directly proportional to gate area and exponentially related to $1/t_{ox}$ [11] indicating that the area of FGTs should be kept minimal. Due to reasons explained earlier in this thesis, it is difficult to analytically predict the J vs. V_{ox} curves described by Eq. 2.4 for a tunneling junction having t_{ox} close to 5 nm. The thinner SiO_2 in the scaled down processes significantly lowers the required energy to move charge carriers through the insulator, so a smaller V_{ox} is needed to initiate F-N tunneling. From Eq. 2.4, the exponential relationship of V_{ox} and t_{ox} with respect to J is apparent. However, the reduction in the tunneling energy naturally comes at the risk of degraded charge retention.

Due to the decreased margin between effective F-N tunneling and junction breakdown voltages, there is also an increased risk for an immediate oxide breakdown. Also the long-term application of a relatively low tunneling voltage (V_{tun}) may cause the oxide to break. This phenomenon is referred to as Time-Dependent Dielectric Breakdown (TDDB) [30]. Fortunately, F-N tunneling is self-limiting. That is, as the charge Q_{fg} at the floating node changes as a result of tunneling, V_{ox} changes accordingly. Thus the current flow in the dielectric decreases and the TDDB effect reduces. However, the TDDB effect may limit the exploitation of continuous programming in adaptive systems. If the terminal voltages of the FGT are biased in such a way

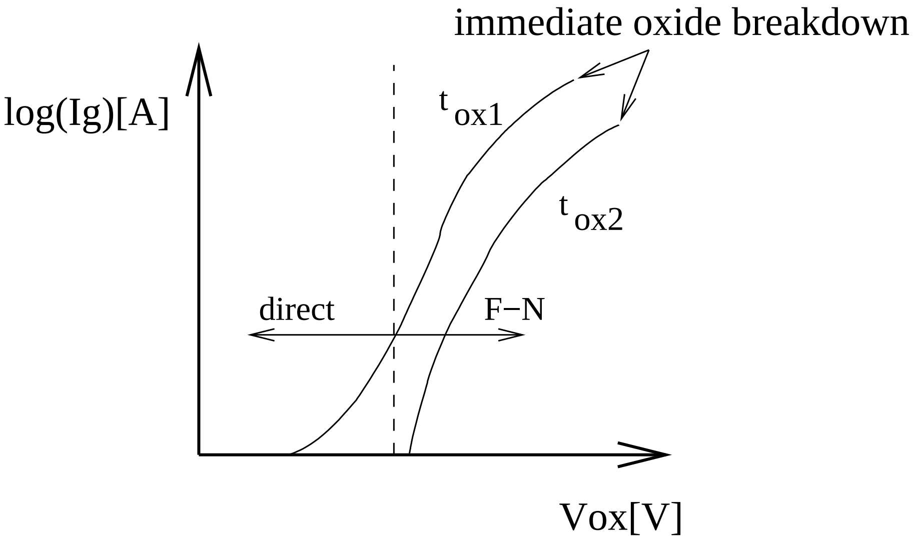


Figure 3.2: An illustration demonstrating the exponential dependence of the gate current I_g on the voltage across the oxide V_{ox}. The oxide thickness t_{ox1} is smaller than t_{ox2}, and thus the device with t_{ox1} produces a measurable gate current below the voltages corresponding to the energy barrier height. In contrast, the direct tunneling is negligible with the device that has thicker oxide (t_{ox2}). For low electric fields across the oxide, the dominant mechanism is direct tunneling and for high fields F-N tunneling respectively.

that the F-N tunneling is constantly competing with the IIHEI process so that gate currents are in equilibrium, the self-limiting mechanism that normally decreases V_{ox} will not prevent dielectric breakdown. Therefore, extra care is required to keep electric fields within reasonable bounds to avoid the TDDB effect in such situations.

Reduced oxide thickness is not the only reason for a leaky floating-gate in DSM CMOS. Another source of leakage is a low quality interlayer dielectric [24, 25]. In addition, the oxide wears out as the tunneling junction is exposed to repeated high field stress. In practice, this long-term effect is caused by trapped charge which increases the current flow through the oxide and may decrease the charge retention ability of the FGT [26]. This phenomenon is more pronounced in DSM processes, because the thin oxide is more vulnerable to changes caused by the trapped charge.

In practice, a tunneling junction can be formed by using an ordinary thick oxide pmos transistor whose gate is connected to the floating-gate, and the drain and the source are tied to the n-well contact, see Fig. 3.3 (a). Such a device forms an n-well capacitor. A triple-well technology is provided in most DSM processes, so in principle p-well capacitors can also be

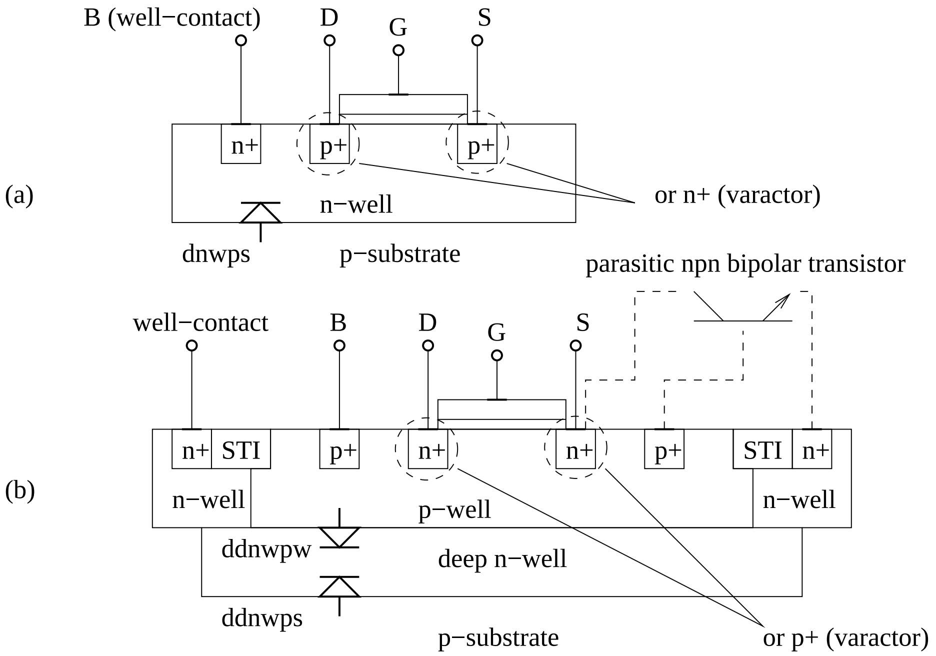


Figure 3.3: Lack of poly-poly capacitors in single-poly process impinges the employment of negative voltages, because well-capacitors must be used in order to avoid (leaky) metallic connections to the floating-gate. (a) The problem with the n-well capacitor is that one of the diodes (dnwps or D/S-n-well) will always be forward biased when either a negative or a positive tunneling voltage is applied, regardless of the doping (n+ or p+) of the drain and the source terminals. (b) The isolated p-well can be pulled below the substrate, but latch-up protection requires that the p-well capacitor is surrounded by a continuous p-well strap so that the adjacent p-well capacitors cannot be connected with the poly-layer. In addition, the condition $V_{n-well} > V_{p-well}$ must be satisfied in all circumstances. Varactors should be used to prevent the occurrence of redundant parasitic bipolar transistors.

used. However, the Design Rule Check (DRC) rules may require continuous well straps (latch-up protection), which prevents poly-layer routing between devices sitting in wells of different types, see Fig. 3.3 (b), preventing this approach.

Some technological limitations must be taken into account when bidirectional tunneling is considered with a DSM single-poly processes. It is challenging to use a well-capacitor to make a tunneling junction that allows bipolar tunneling voltages due to the following reasons: 1) Assuming the drain and the source terminals in Fig. 3.3 (a) are doped with $p+$ and tied to $n+$ n-well contact, applying a positive V_{tun} to tunneling junction is OK, but the n-well-p-substrate diode (dnwps) will be forward biased if the n-well is pulled more than the threshold voltage of the diode below the ground. 2) If the n-well contact is separated from the drain and source and set to ground, applying a negative V_{tun} to tunneling junction (D and S terminals) is OK, but a positive V_{tun} causes the D-n-well and S-n-well diodes to be forward biased. 3) If a varactor configuration is used, i.e. the drain and the source terminals are doped with $n+$, the n-well contact can not be separated from the drain and source terminals. In this case, the n-well-p-substrate diode (dnwps) will be forward biased when applying a negative V_{tun} to the tunneling junction. This leaves two choices: either to make two separate tunneling junctions, one for positive and one for negative V_{tun}, or apply the approach described in Fig. 2.5, which allows bidirectional tunneling using positive voltages only.

Another concern arises from the high voltages needed for F-N tunneling. Realization of a floating switch[1] that can withstand adequate voltage levels in a DSM CMOS is generally difficult. It is possible to construct high voltage switches from devices having lower voltage tolerances [32], but this approach results in very complex switching circuits, which consume a lot of area.

In conclusion, F-N tunneling is feasible in standard single-poly DSM CMOS processes. However, practical implementations of selective F-N tunneling tend to be more complex than in the earlier technology nodes because ensuring a robust operation requires ever more complex DRC rules and technological limitations. Also, the direct tunneling is no longer insignificant, which shortens the retention time of the FGT. However, the direct tunneling can in some cases be taken advantage of, as was suggested in [44]. According to the hypothesis made in [44], the adaptive properties of an FGT are directly available with thin-oxide devices in a general purpose DSM process. This is based on the assumption that when the oxide thickness is decreased to approximately 2 nm, F-N tunneling is replaced by direct tunneling, which in turn enables AFGA-like behavior (see Section 4.3 and Fig. 4.14 for the explanation) even within the nominal process voltages. This approach is examined further in Section 4.3 with measured results from a prototype AFGA circuit.

As mentioned in the previous chapter, the IIHEI mechanism is the preferred choice for programming FGTs in DSM processes. The typical programming scheme for an array of p-type FGTs uses the IIHEI mechanism for accurate local programming of devices and F-N tunneling for global erasure. The major advantage of using global erasure is that high voltage tolerant switches needed for local F-N tunneling are avoided. Accurate programming with IIHEI requires device-specific characterization of the FGT to calculate

[1] A non-floating switch is either a pull-up or a pull-down device. A floating switch connects two nodes from which neither of them is a supply voltage (comparable to pass transistor used in digital circuits).

bias dependent values for both fields in order to achieve a given target current with a minimal amount of injection cycles. Actually, it should be possible to hit the target in one injection cycle if the characterization is done carefully. However, this becomes impractical when the number of FGTs on the chip is large. Also, in order to maintain accuracy which enables one shot programming, the characterization should possibly be repeated periodically to compensate for the long-term changes in the oxide. Hence, in practice an iterative approach is used for accurate programming of large FGT arrays. This is important, because if the target is overshot, the programming must be started over and all FGTs erased. A method of realizing such a scheme is presented in [38].

Also IIHEI is feasible in standard single-poly DSM CMOS processes. There are no special issues that need to be addressed when considering the use of IIHEI in a DSM process as compared to the technology nodes having $t_{ox} > 6$ nm. Of course, the injection parameters must be scaled according to the technological parameters, but the same circuit topologies can be used in this respect.

If available thick oxide devices are used to maximize retention time, the migration to DSM technologies does not enable any new methods for programming FGT. However, if the charge retention is of secondary interest, which could be the case for example in the adaptive systems, mechanisms like direct electron tunneling and the hole tunneling can be utilized with ultra-thin oxide devices.

Chapter 4

Experimental results

In the previous chapter, it was claimed that FGTs are feasible in DSM standard CMOS. Several test chips were designed and their functionality was tested through measurements to confirm this claim. Tested process nodes range from 0.18 μm to 65 nm, each prototype chip includes different FGT circuits with varying complexity.

4.1 Test Chips 1 and 2

The first objectives were 1) to verify the functionality of both F-N tunneling and IIHEI in DSM CMOS and 2) to roughly evaluate the charge retention capabilities of FGTs made of thin oxide devices below the 0.25 μm technology node. For this purpose, two chips were designed and manufactured, one in 0.18 μm CMOS and the other in 130 nm CMOS [47].

4.1.1 FGT Implementation in 0.18 μm CMOS

Fig. 4.1 shows the schematic of the designed circuit. All transistors are "thin" oxide devices. For this technology node, this means a nominal oxide thickness of approximately 4 nm, so that only moderate retention times can be expected. The programming scheme is indirect and the circuit uses F-N tunneling for global erasure and IIHEI for local programming.

The circuit comprises a programming transistor $M1$, signal transistors $M2$ and $M3$, an input capacitor Min and a tunneling capacitor Mt. The capacitors are implemented with PMOS transistors with their gate terminals connected to the floating node and all other terminals (drain, source and bulk) tied together. $M0$ is the diode connected input transistor of the current mirror, and $M4$ can be used as a reference for $M2$ (identical dimensions). In addition, there are switches for choosing the desired output device ($c1$- $c3$), and for controlling the programming-measurement cycle (sw and

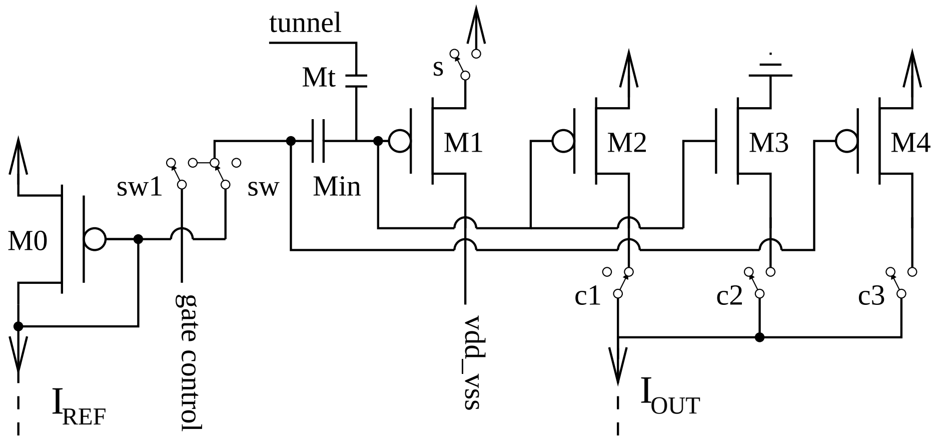


Figure 4.1: A schematic illustration of the FGT circuit, which was implemented to test the feasibility of F-N tunneling and IIHEI in 0.18 μm CMOS.

*sw*1). The purpose of switch s is to isolate FGTs during injection. Terminals denoted as *tunnel* and *vdd_vss* are connected to unprotected analog IO-pads which permit the use of voltage levels exceeding the nominal supply rails (1.8 V and 0 V).

The prototype chip includes 16 different versions of the same FGT circuit. These are all connected to the same output node and also have $M0$ and $M4$ in common. The schematics of the different versions are exactly the same excluding the differences in dimensions of Min, Mt and $M1$. Different device sizes were implemented to examine the effect of different capacitive ratios on the effective threshold voltage shift. The use of n-diffusion, instead of p-diffusion in the drain and source regions was tested to find out if this would work for Mt, and if the removal of the LDD layer [1] from $M3$ would have any effect on IIHEI.

The output current I_{out} was converted to voltage V_{out} with a 1 MΩ resistor. The probe of an oscilloscope was also connected to the output node. I_{REF} was biased by connecting the input node to ground through a 500 kΩ resistor. It was expected that the charge retention would be quite poor due to the quite thin gate oxide (approx. 4 nm) therefore; the focus was on examining the charge retention and testing the F-N tunneling in different configurations.

An example of how the output voltage changes as a function of tunneling time can be seen in Fig. 4.2. In this measurement, the supply voltage was set to 1.8 V, drain of $M1$ was grounded, $M2$ was connected to the output node, and a pulse train consisting of 4 V tunneling pulses was applied to

[1] Lightly doped drain (LDD) prevents CHE, which is normally considered harmful.

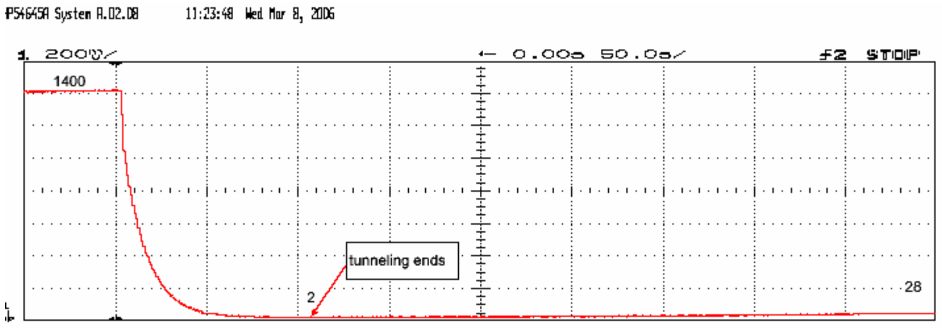


Figure 4.2: The output voltage V_{out} driven by PMOS FGT ($M2$) decreases in response to applied tunneling voltage (effectively a gate voltage sweep). The output voltage starts slowly to increase due to leakage, after the tunneling node is grounded.

Device	$M0$	$M1$	$M2$	$M3$	$M4$	Min	Mt
W/L	2/2	1/0.5	2/2	0.5/0.5	2/2	10/10	1/1

Table 4.1: W/L ratios of transistors used in measurement shown in Fig. 4.2

Mt. The W/L ratios of the transistors are shown in table 4.1. The period of a single tunneling pulse was 2 ms, and the pulse was repeated 50000 times resulting in a total tunneling time of 100 s. The required tunneling time was quite long, because V_{tun} was only 4 V in order not to break down the devices. Also, Min was made sufficiently large, so that V_{fg} was a strong function of input voltage. Consequently, the floating-gate capacitance C_{tot} is quite large. The corresponding output voltages in the beginning, after tunneling and 350 s after the end of the tunneling cycle are marked in the figure in millivolts.

The tunneling test effectively replicates the v-i response of a PMOS transistor for a gate voltage sweep. However, now there are two phenomena that are superimposed, and both contribute to the slope of the curve seen in Fig. 4.2. These are: 1) the biasing of $M2$ changes as the V_{fg} increases (as in normal gate voltage sweep), and 2) the gate current I_G per tunneling pulse decreases as V_{fg} approaches V_{tun}. Before tunneling, $M2$ delivers the maximal current, because V_{fg} is effectively grounded. As the tunneling begins, the V_{fg} changes rapidly because the difference between V_{tun} and V_{fg} is large. As the potential difference gets smaller, the rate of change in effective V_{th} gets slower. Finally $M2$ drops to the linear region and the

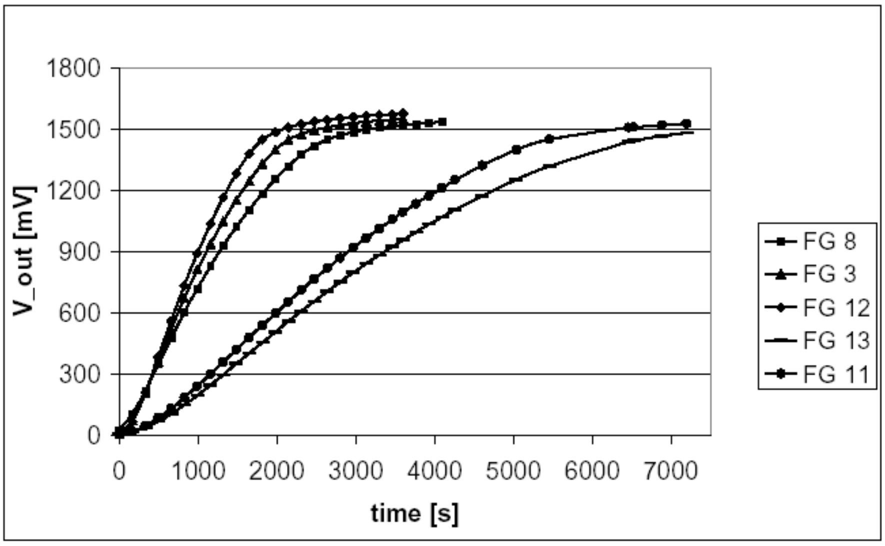


Figure 4.3: The output voltage of a PMOS FGT ($M2$) increases due to direct tunneling when first tunneled in cut-off and then the V_{tun} is grounded. The W/L ratio of Min is 10/10 for F11 and F13 and 5/5 for the rest. The rate of change in V_{fg} is proportional to C_{tot} as can be expected.

effective V_{th} hardly changes. It can be seen that the output voltage starts to increase immediately after V_{tun} is grounded. This was expected, because the thin gate oxide (approx. 4 nm) inevitably results in poor charge retention.

When the intentional tunneling is finished, electrons return to the floating-gate due to the unintentional direct tunneling. In consequence, the initial output voltage is slowly restored, as is shown in Fig. 4.3. The rate of change in V_{fg} is proportional to C_{tot}, as can be expected. The curves obtained from FG11 and FG13 have Min of size 10/10 and the rest of 5/5, so that the responses are clearly different. Furthermore, by quadrupling the area of Min, the time needed to achieve an equilibrium state is approximatively tripled, which is consistent with the increase in C_{tot}. The rest of the differences are explained with differences in the sizes of Mt and $M1$. It should be noted that the time scale in Figure 4.3 is two hours.

It was found out, that the tunneling capacitors with n+ diffusion were not functional. The reason was that they were not processed correctly because of a missing marker layer, which indicates that they are supposed to be varactors. Because of interfering direct tunneling caused by the thin oxide, it was challenging to reliably measure the contribution of the IIHEI on the injected charge. That is why, these results are not reported. The effect of the missing LDD-layer in part of the FGTs could not be tested for the same reason.

4.1.2 FGT Implementation in 130 nm CMOS

A simplified version of the test chip 1 was designed and manufactured in a 130 nm CMOS, because the IIHEI programming could not be properly verified in a 0.18 μm CMOS. Therefore, 3.3 V “thick” oxide IO-transistors were used to minimize the leakage due to direct tunneling. These transistors have t_{ox} of approx. 7 nm, which is comparable or even thicker than that of thin oxide devices in a typical 0.25 μm technology. As FGTs implemented in a 0.25 μm CMOS are reported to have retention times over 10 years [48], it could be safely assumed that the implemented FGTs would not experience considerable charge loss. In the unlikely case they turned out to be leaky despite the thick gate oxide, it would indicate the existence of some other leakage source that presents itself only when migrating FGTs to DSM CMOS processes.

The schematic of the designed circuit is shown in Fig. 4.4. An indirect programming scheme allows global tunneling erasure and local IIHEI programming with minimal amount of additional switches. There are two copies of the presented circuit in parallel, but only one of the FGTs has an additional switch s (ordinary PMOS transistor), otherwise they are identical. The switch was added to examine if an acceptable level of selectivity could be achieved with a simple PMOS switch. All terminals ($Vtun$, $Vinj$, Vin, I_{out} and Vdd) are shared between these two FGTs and connected to unprotected analog IO-pads, so that their magnitude is not limited to nominal supply voltages. Both FGTs have a readout switch $c1$ of their own that can be controlled independently for current readout.

The main purpose of this test chip was to verify the functionality of IIHEI in DSM CMOS. Both F-N tunneling and IIHEI work fine and the charge retention time was verified to be at least several days. It was observed that a voltage of at least about 7 V is required to erase the charge by tunneling. This result is consistent with the estimated minimum field 6.4 x 10^8 V/m, since 7 V per 7 nm gives an initial field of 10^9 V/m.

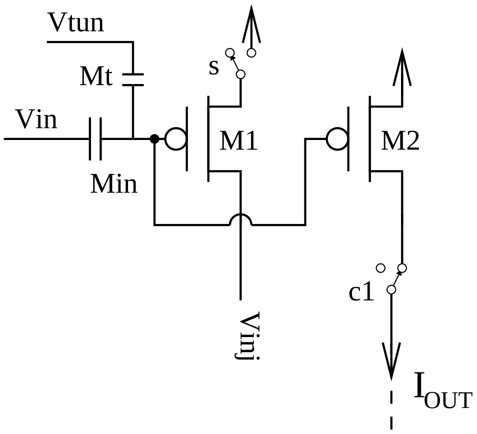


Figure 4.4: A schematic of the FGT circuit implemented in a 130nm CMOS.

4.2 An FGT Current Source Implemented in a 90 nm CMOS

One component that greatly improves the flexibility of analog design is a tunable voltage source or current source/sink. It is very useful for generating on-chip bias currents or voltages in analog systems and references in both analog and digital systems. The need for accurate low-voltage or low-current references is emphasized in low-power applications. An FGT based temperature compensated programmable voltage reference is presented in [39]. The same techniques can be applied to provide a current source or sink. FGT based references are well suited for low supply voltage operation.

The possibility of turning the continuously decreasing intrinsic accuracy of minimum sized devices to a useful property was examined in [45]. The idea of this study was to get the current sink to be automatically calibrated within 1 % of the nominal value at the 4 σ confidence interval for input currents ranging from 1 to 10 μA. In this calibration system a low accuracy current mirror provides 80 % of the targeted output current and a subset of output transistors of the current mirror (minimum sized devices) the remaining 20 %. A counter goes through all possible combinations of transistors connected to the current mirror until the output current hits the target (is between the two programmable limits). In order to test the concept in practice, two pieces of simple FGT current references were included in the design to provide the programmable limits, as presented in Fig. 4.5. A prototype chip was implemented and manufactured in a 90 nm CMOS.

The two FGT current sources were implemented with double oxide devices allowing the programmed charge to be stored on the floating gate. The programmed state of the FGT references can be examined independently by

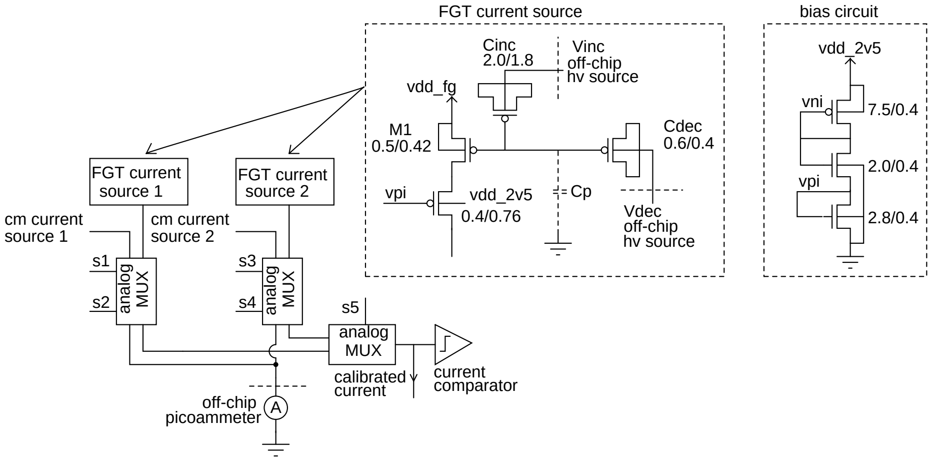


Figure 4.5: The principle of checking if the calibrated current is between the limiting references. FGT technology allows both bounds to be programmed independently within designed current range with an arbitrary precision. The upper bound and the lower bound are checked one after the other with a current comparator. An off-chip picoammeter is used to observe the effective change in the drain current as a response to the programmed charge in the floating gate. This design uses off-chip high voltage sources for controlling the programming voltages (bidirectional F-N tunneling). Cascode transistor stabilizes the drain voltage of $M1$, and Cp illustrates the parasitic capacitance. Also shown is how the the bias voltages for the precision comparator and the cascode transistor are created.

an off-chip picoammeter. The current read-out allows the comparison of the acquired data with simulated curves.' The targeted current range in this application was $1 - 10\mu A$, but there are no fundamental restrictions on using subthreshold currents. Certainly, off-chip current measurements are more challenging with smaller current levels due to the noise.

The two FGT current sources in Fig. 4.5 can be programmed independently. Analog MUXes pass the reference current either to the current comparator or to an off-chip picoammeter. Additional nodes, *cm current source x* in Fig. 4.5, are connections to current mirror references just for backup, in the case the FGT sources proved to be inoperable. The system works as follows: First the FGT current sources are programmed to the target with help of the off-chip picoammeter. These two FGT current sources set the lower and upper limits for the current to be calibrated. In the run-mode, the comparator sequentially checks if the *calibratedcurrent* is between these limits (on target).

Programming of the FGT current sources is realized with bidirectional F-N tunneling to keep the system as simple as possible. The programming

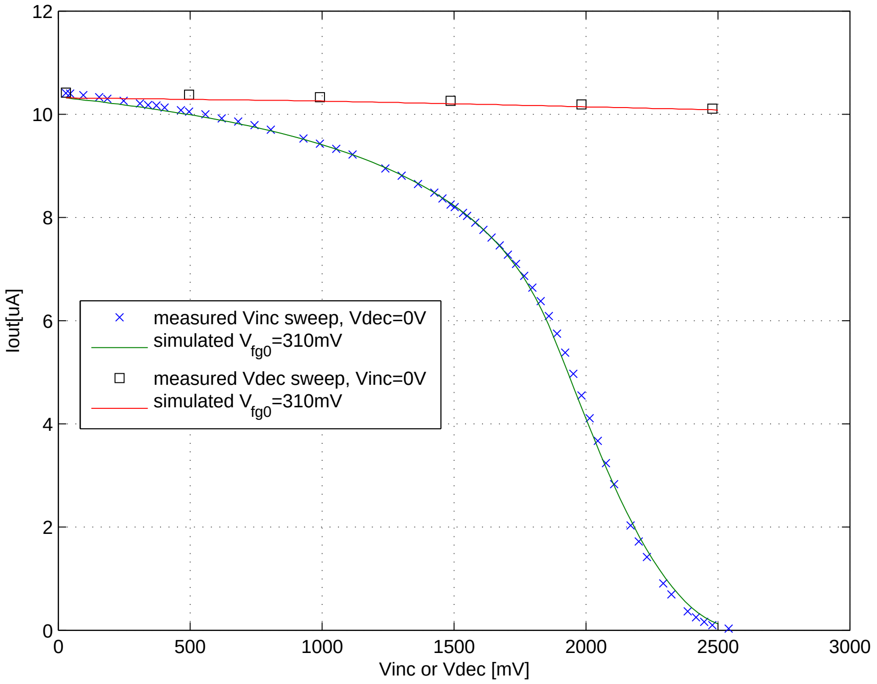


Figure 4.6: Measured IV curves for both capacitors $Cdec$ and $Cinc$ with fits obtained from the circuit simulator assuming: Cp=1.49fF, V_{fg0}=310mV and vdd_fg=2.44V. $Cdec$ is grounded while sweeping $Cinc$ and vice versa. The $\frac{Cinc}{Cdec}$ ratio is large enough so that V_{fg} is almost independent of the voltage applied to $Cdec$, enabling F-N tunneling through $Cdec$ in both directions.

scheme is explained in Fig. 2.5. In this case $Cinc$ is the control capacitor while $Cdec$ forms the tunneling junction, thus $Cinc > Cdec$. What follows is that by applying a high positive voltage ($V_{inc} = V_{tun}$) to $Cinc$ and grounding $Cdec$ ($V_{dec} = 0$), electrons are added to the floating gate and the drain current of $M1$ increases. Respectively, by applying a high positive voltage to $Cdec$ and grounding $Cinc$ the effect is reversed. The $\frac{Cinc}{Cdec}$ ratio must be high enough to achieve a sufficient potential difference across $Cdec$ in both directions. In this design the area of $Cinc$ is 15 times larger than the area of $Cdec$. Consequently, the floating gate voltage V_{fg} is a strong function of V_{inc} as can be seen from Fig. 4.6.

A cascode transistor is used to stabilize the drain voltage of $M1$. Besides minimizing the capacitive coupling of disturbances, for example, due to the switching of analog MUXes via C_{gd} of $M1$, it also minimizes the IIHEI effect in $M1$.

4.2.1 Measured data

In the following, the acquired data obtained from the prototype chip is examined and this data is used to determine correct simulation parameters for the circuit simulator to get matching responses. All measurements are performed with DC signals so that the influence of the load capacitance (IO-pads and off-chip capacitances) is negligible. The output current, $Iout$, of the FGT current source is a function of many parameters, such as the power supply voltage vdd_fg, the programmed V_{fg0} and the parasitic capacitance Cp. The correct combination of simulation parameters can be found by tying the $Cdec$ and $Cinc$ together and then sweeping the input in small steps in order to extract the IV curve of the FGT current source. By changing the charge on the floating gate between consecutive sweeps a family of IV-curves is provided. This family of IV curves can be fitted to the data obtained from the circuit simulator by assuming that only the initial V_{fg0} (programmed voltage when $Vdec = Vinc = 0$V) changes and everything else remains fixed. Only a small family of IV curves is needed to get pretty accurate estimates for the actual vdd_fg, Cp and V_{fg0} seen by the FGT.

A pulse train consisting of 0.1ms pulses with amplitudes ranging from 5 V to 6.5 V (depending on the operation point and direction of charge transfer) was used for programming the FGT current source. Moderate amplitude programming pulses were used in conjunction with a rather large amount of pulses to get a fine resolution programming accuracy and to avoid excessive stress on the gate oxide. About 33,000 consecutive programming pulses were required in order to get a 10 nA change in $Iout$ if the initial $Iout$ was 10 μA. Note that the effective tunneling voltage V_{ox} is not equal to the applied tunneling pulses, but depends on the initial charge Q_{fg}: in contrast to constant charge injection presented in Section 4.4, the gate current in electron tunneling always depends on the initial charge Q_{fg} and consequently the tunneling amplitude must be adjusted according to the operating point.

Fig. 4.7 shows a family of measured IV curves with associated data obtained from the circuit simulator. First the IV curves are shifted left from the initial condition because electrons are removed from the floating-gate. Then the curves are shifted right by adding electrons to the floating-gate. With the assumptions made above the circuit simulations suggest a parasitic capacitance of 1.49 fF and vdd_fg to be 2.44 V. Programmed floating-gate voltages V_{fg0} after each programming cycle can be estimated by giving V_{fg0} as an initial condition for the circuit simulator and finding the best match for the measured IV-curve. These estimates of V_{fg0} are shown in the legend. A select set of IV curves presented in Fig. 4.7 (from initial to after tun3 down) are shown in the semilog scale in Fig. 4.8 to illustrate that the subthreshold region can be tracked very well with the simulator by using the proposed method.

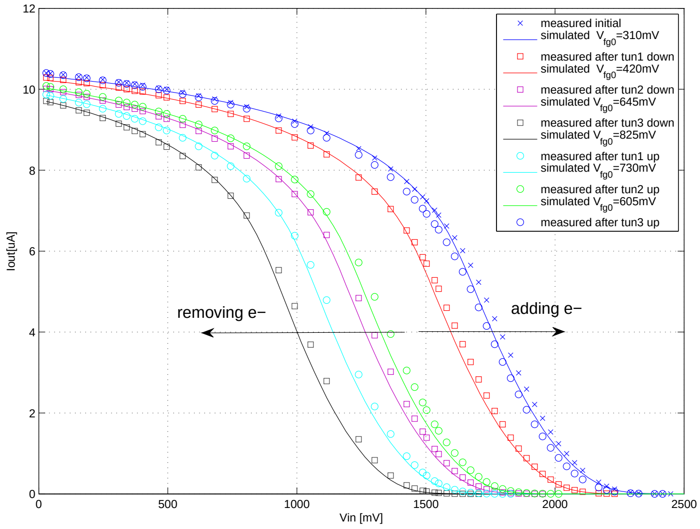


Figure 4.7: The measured IV curves of the FGT current source with the associated data obtained from the circuit simulator. Vin is applied to both capacitors $Cdec$ and $Cinc$ simultaneously and V_{fg0} is the estimated floating gate voltage after programming when $Vdec = Vinc = 0$V. Estimated parasitic capacitance and power supply voltage from the fits are: Cp=1.49 fF and vdd_fg=2.44 V.

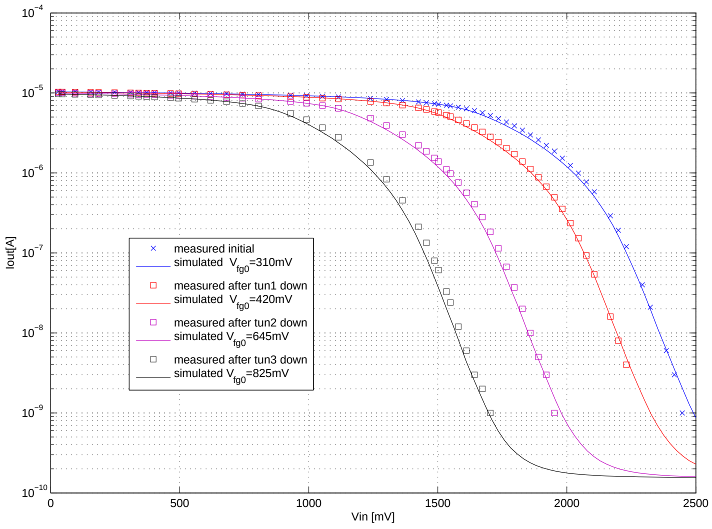


Figure 4.8: A set of IV curves presented in Fig. 4.7 (from initial to after tun3 down) on the semilog scale with the associated data obtained from the circuit simulator. The semilog scale shows how the low current end agrees well with the circuit simulator.

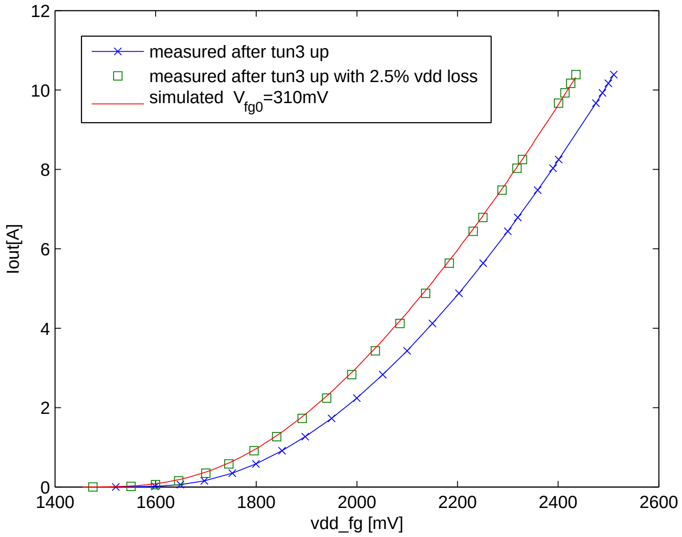


Figure 4.9: Measured $Iout$ as a function of vdd_fg. The difference in simulated and measured data suggests a 2.5 % voltage drop in vdd_fg line (Cp=1.49 fF, V_{fg0}=310 mV).

The measurements of Fig. 4.7 seem to suggest a large voltage drop in the power supply line. The voltage drop experienced by the vdd_fg line was examined more carefully by plotting the $Iout$ as a function of vdd_fg. Fig. 4.9 is obtained by tying the capacitive inputs to ground ($Vdec = Vinc = 0$ V) and taking one initial condition (V_{fg0}=310 mV) for the FGT and then plotting the $Iout$ as a function of vdd_fg. The simulated response coincides with the measured data when the measured data is corrected with a 2.5 % voltage drop (Cp is 1.49 fF in simulation). This number includes the on-chip interconnects, the flip chip package, the off-chip PCB lines as well as the device mismatch. However, even the joint effect of the device mismatch and resistance in the power line does not explain such a big difference in the simulated and measured curves. The conclusion is that the circuit simulator simply fails to model the used devices precisely. This suspicion is further supported by the fact that the used models are labeled as “pre-production” maturity status.

A parasitic capacitance of 1.49 fF is consistently assumed in all circuit simulations which matches well with the parasitic capacitance extracted from the layout 1.37 fF (RC-typical extraction model). The effect of parasitic capacitance on the IV response is shown in Fig. 4.10. The figure demonstrates the importance of taking into account the effect of the Cp in the design phase. Failing to do so can lead to considerable bias offset

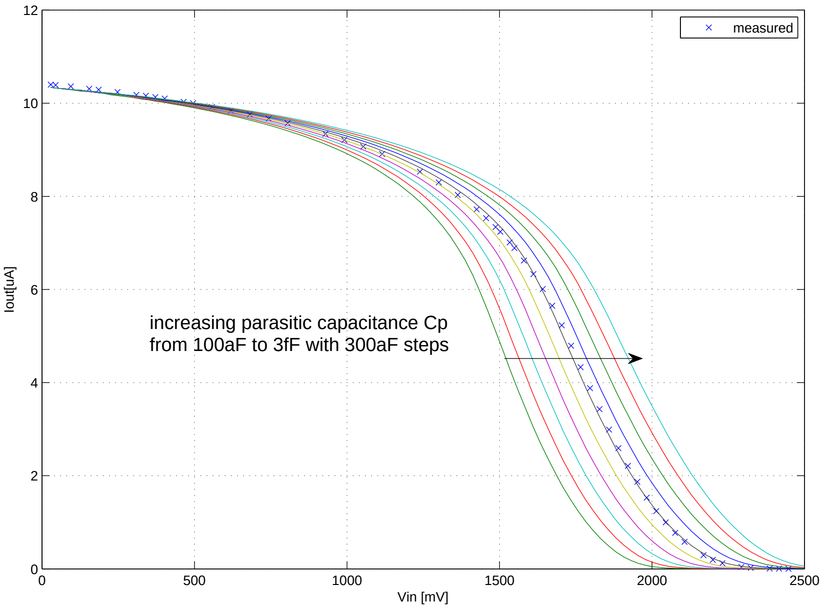


Figure 4.10: The simulated effect of the parasitic capacitance Cp on $Iout$ (V_{fg0}=310 mV, vdd_fg=2.44 V).

depending on the operating point of the FGT (∼250 mV @ 5 μA in this example).

The effect of the vdd_fg on the $Iout$ is illustrated in Fig. 4.11 which plots the simulated IV curves against the voltage drop in the vdd_fg line (Cp=1.49 fF in simulations).

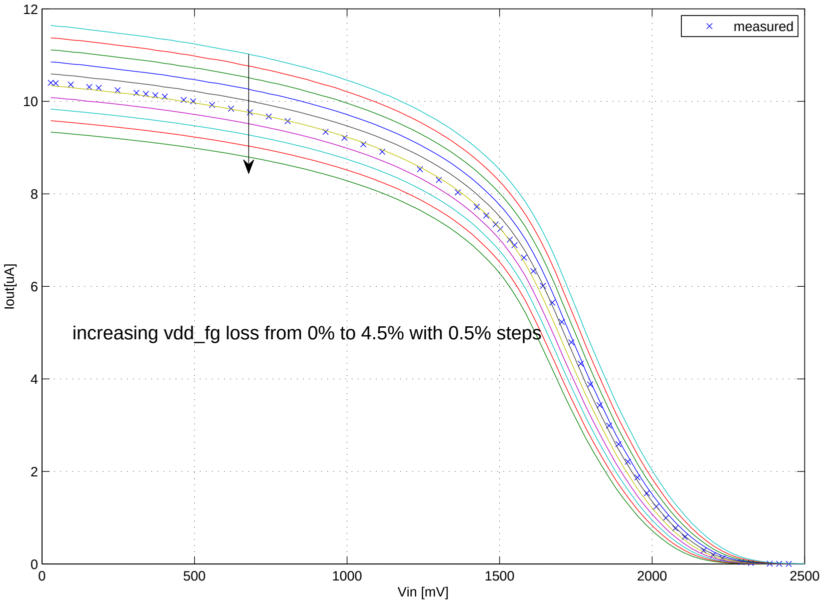


Figure 4.11: The simulated effect of the voltage drop in *vdd_fg* line on $Iout$ (Cp=1.49 fF, V_{fg0}=310 mV).

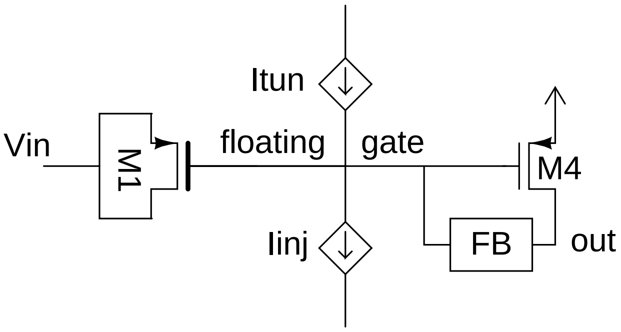


Figure 4.12: Due to the feedback loop, complementary F-N tunneling and IIHEI are continuously trying to restore the floating-gate voltage to a new equilibrium state on a slow timescale and thus the circuit implements an amplifier with a built-in high-pass filter.

4.3 A Thin Oxide AFGA Implemented in a 65 nm CMOS

The Autozeroing Floating-gate Amplifier (AFGA) amplifier continuously tries to return to a steady-state value on a slow timescale and thus behaves like a high-pass filter. This autozeroing feature is based on the complementary F-N tunneling and IIHEI injection mechanisms that modify the floating-gate charge. As soon as the steady-state value of the amplifier's output is disrupted by the applied input signal, either the injection current exceeds tunneling current or the other way around, until a new equilibrium state is reached (see Fig. 4.12). Due to small gate currents the floating-gate charge adapts only at a slow timescale. Thus, for input frequencies exceeding the time constants allowed by the gate currents, the floating-gate is held nearly fixed and the AFGA behaves like an ordinary amplifier. While the tunneling and injection currents set the low-frequency cutoff, the high-frequency cutoff is independently set by the bias current through $M4$, so that the AFGA has a bandpass transfer function.

Inspired by the predictions made in [44], a prototype of AFGA made of thin oxide devices was designed and manufactured in a 65nm CMOS process. The schematic of the design (presented in Fig. 4.13) does not differ from the original [17] except for the fact that the transistors, $M2, M4, M6$, are thin oxide devices. Although it would be possible to use thin oxide devices for all transistors, thick oxide devices were used for capacitively coupled transistors (excluding the tunneling junction). This is because of geometrical limitations of digital transistors (L<180 nm) [2] and better predictability of

[2] Geometrical limitations apply to the design kit used but not in general.

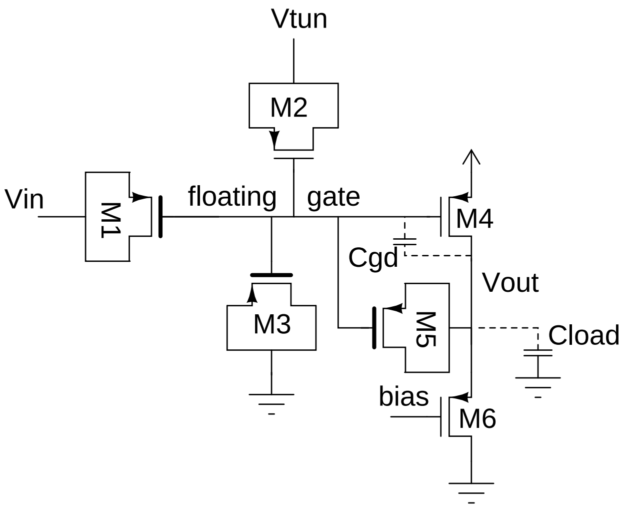


Figure 4.13: A schematic of an AFGA comprises both thick (bold gate) and thin oxide p-type transistors. No voltages exceeding the nominal supply voltage of the design kit are needed for achieving the autozeroing feature.

functionality. The prototype chip contains two versions of the AFGA, one made of the general purpose digital transistors (psvtgp) and the other of low power digital (psvtlp) transistors. The thick oxide devices (psvt25) and dimensions of the transistors are identical for both versions. All thin oxide devices were drawn using minimum dimensions allowed by the design rules (W/L=0.135 μm/0.06 μm). The W/L ratios of thick oxide devices were selected to be (all dimensions given in μm) 4/1 for $M1$ 0.4/0.4 for $M3$ and 0.5/0.5 for $M5$ respectively.

The aim of this design was to achieve a small gain (∼10 dB) amplifier with a bandwidth in the kHz range. The explanation for using a small gain is that there is a gain-bandwidth tradeoff in the AFGA and small gain simply sets the bandwidth on a convenient region so that it can be easily measured with the equipment available. The psvtgp and psvtlp targeted for different purposes actually have surprisingly large differences. For example, the gate oxide is 30 % thicker in low-power devices to minimize leakage currents. Also the nominal supply voltage is higher for the low-power devices: 1.2 V for lp and 1.0 V for the gp respectively. These differences become visible when the bandwidth of the amplifier is measured. The minimum-sized devices also express considerable mismatch.

In the following, a detailed transistor level operation of the AFGA is explained with help of Fig. 4.13. The maximum gain of the amplifier depends on the open loop gain set by the transconductance of $M4$ g_{mM4} and bias current through $M6$. Transistor $M5$ creates a capacitive feedback loop which affects the closed loop gain of the amplifier. Assuming a large open loop gain, the feedback keeps the floating-gate voltage virtually constant in the passband. In that case, the closed-loop ac gain of the amplifier in the passband is adjusted by the C_{M1}/C_{M5} ratio [17] (see Eq. 4.1), which is identical in both versions. Naturally, the parasitic capacitances (mostly floating-gate-to-drain overlap capacitance) decrease the gain. Eq. 4.1 changes into Eq. 4.2, when the effect of C_{gdM4} is taken into account.

$$V_{out} = -\frac{C_{M1}}{C_{M5}} V_{in} + V_{fg} \tag{4.1}$$

$$V_{out} = -\frac{C_{M1}}{C_{M5} + C_{gdM4}} V_{in} + V_{fg} \tag{4.2}$$

It is possible to further simplify the circuit by omitting the transistors $M2$ and $M3$, since a separate tunneling junction ($M2$) is not really needed for thin oxide devices and $M3$ is just for linearization improvement. However, these devices were included in the design, because it was not clear if the behavior suggested by the circuit simulator and the models were trustworthy for this application. $M2$ also offers extra flexibility for testing the voltage dependency of the direct tunneling and the capacitor represented by $M3$ exists anyway because of parasitics, even if not explicitly drawn.

Fig. 4.14 presents the response of the lp-version. It is evident from Fig. 4.14 that qualitatively the behavior of the designed AFGA resembles the original design. It amplifies the input signal, and adapts to changes in the dc-level in about 1.5 seconds. The response of the gp-version is similar but the output returns to the steady-state in few hundred μs. The bandwidth of the AFGA for both versions is shown in Fig. 4.15 for different above threshold bias conditions and tunneling voltages ($Vtun$). Measurements were performed by coupling a signal generator to the input and then observing the output voltage of the AFGA with an oscilloscope. A sine wave signal with relatively high peak-to-peak voltage (100 mV) was used for improved accuracy, because the gain of the amplifier is quite small and the gain had to be estimated from the screen of the oscilloscope. The test probe of the oscilloscope was set to 10x in order to minimize the capacitive loading of the circuit. That is, only the parasitic capacitances, a coaxial cable, a probe and the input capacitance of the scope were loading the circuit, resulting in a capacitive load on the order of 15 pF. The load capacitance does not affect the ac gain in the pass-band, but directly affects the low-pass corner frequency of the amplifier. Hence, doubling the load capacitance approximately halves the 3 dB low-pass corner.

The gain of the gp-version is lower than that of the lp-version as can be expected due to the thicker gate oxide of lp-transistors. Thicker gate oxide results in a lower C_{gdM4} and tunneling conductance (g_{tunnel}) in parallel with

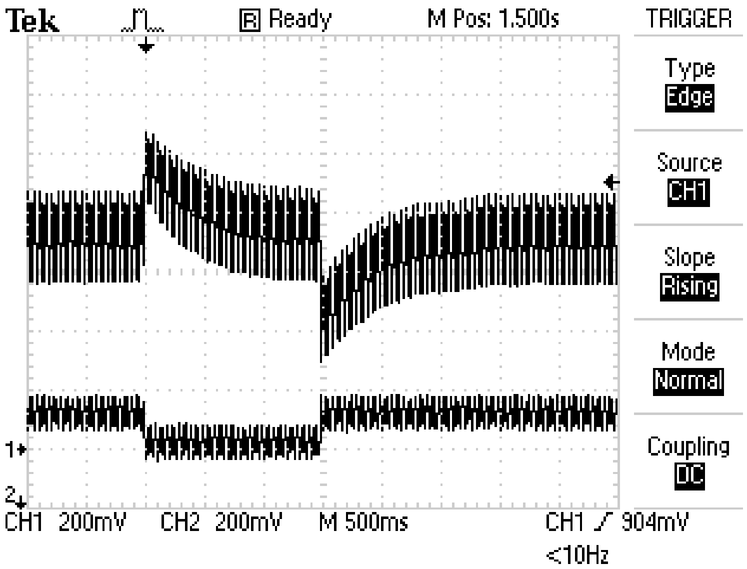


Figure 4.14: The intrinsic autozeroing feature of the AFGA. The 20Hz sine wave input signal (the lower signal) is amplified (the upper signal), but the changes in the dc-level are compensated for in a slow timescale.

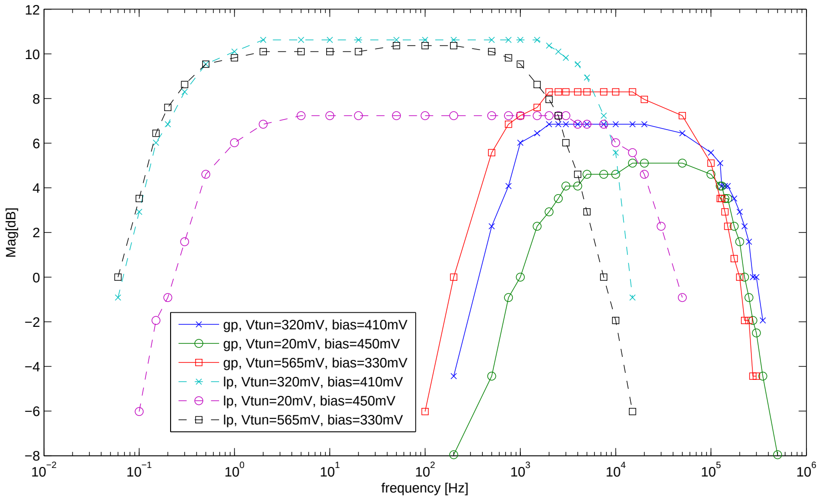


Figure 4.15: The bandwidths of the AFGAs (gp and lp) under three different bias conditions (above the threshold) and tunneling voltages. For the lp-version, the maximum gain is 3.3 (10.4 dB), bandwidth (-6 dB) roughly from 0.4 Hz to 40 kHz, and center frequency from 20 Hz to 100 Hz. For the gp-version, the maximum gain 2.6 (8.3 dB), bandwidth (-6 dB) roughly from 150 Hz to 280 kHz and center frequency from 6.7 kHz to 13.7 kHz respectively.

the input capacitance of $M2$ (and $M4$). The high-pass corner frequencies of the gp-version are also roughly 10^3 higher than those of the lp-version due to the same reason.

For low signal frequencies, the input impedance of the transistor is resistive rather than capacitive. It is possible to define a frequency f_{gate} for which the transformation of input impedance from resistive to capacitive happens [11].

$$f_{gate} = \frac{g_{tunnel}}{2\pi Cin} \tag{4.3}$$

The same reference also gives another equation for the f_{gate} which is based on empirical measurements. For the PMOS transistor the fit based on empirical data is:

$$f_{gate_emp} = 0.5 \cdot 10^{16} \cdot v_{gs}^2 e^{t_{ox}(v_{gs}-13.6)} \tag{4.4}$$

This equation allows us to compute the ratio of f_{gate} frequencies for the two versions of the AFGA. By substituting the nominal gate oxide thicknesses for the lp and gp transistors and assuming that the v_{gs} voltage is the same, we get a ratio of approximately 10^3, which is consistent with the observed differences in the high-pass corner frequencies. Therefore, it is challenging to design a good amplifier for the subkHz range using transistors with very thin gate oxide (svtgp), if standard amplifier topologies are considered. On the other hand, the effect of load and parasitic capacitances dominates the high-frequency inputs. Thus, the differences in the low-pass corner frequencies are in the order of one decade only.

The measured results were compared with the simulated results to find out the circuit simulator's ability to model the response of the AFGA. Fig. 4.16 shows that the measured ac response of the AFGA (gp) matches fairly well to the response suggested by the circuit simulator. In this case, the parasitic capacitances and resistances are extracted from the layout and included in the simulation to get a more realistic response. The simulated amplifier has somewhat wider bandwidth. The lower low-pass corner in the measured AFGA suggests that the actual load capacitance was a bit larger than the estimated 15 pF (no explicit load). The low-pass corner frequency is linearly dependent on the load capacitance. Thus, the circuit simulator is quite trustworthy and models very well the ac behavior of the semi-floating gate amplifier.

The adaptation rate of the AFGA can be examined by considering the step response of the amplifier. A 100 mHz square wave with amplitude of 100 mV was used for this purpose and settling times back to the steady state were recorded under different bias conditions. A step response characteristic of an autozeroing amplifier is shown in Fig. 4.17. Immediately after the edge of the input signal, the output is given by the magnitude of the input step

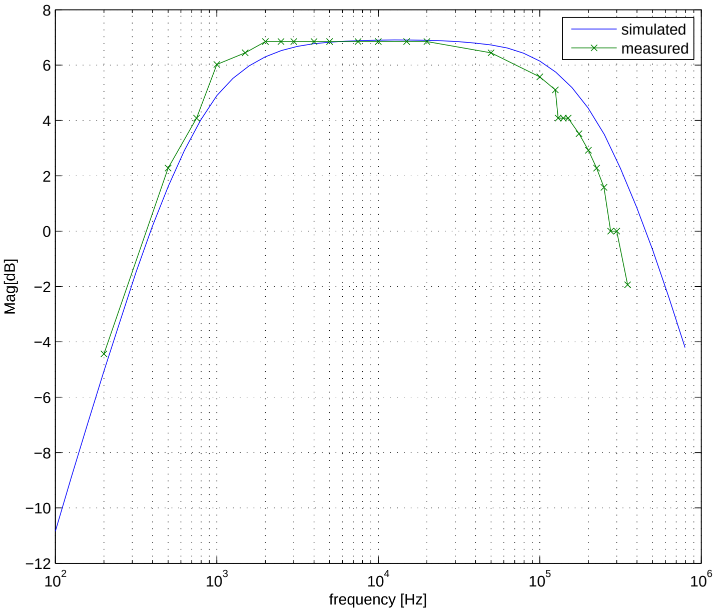
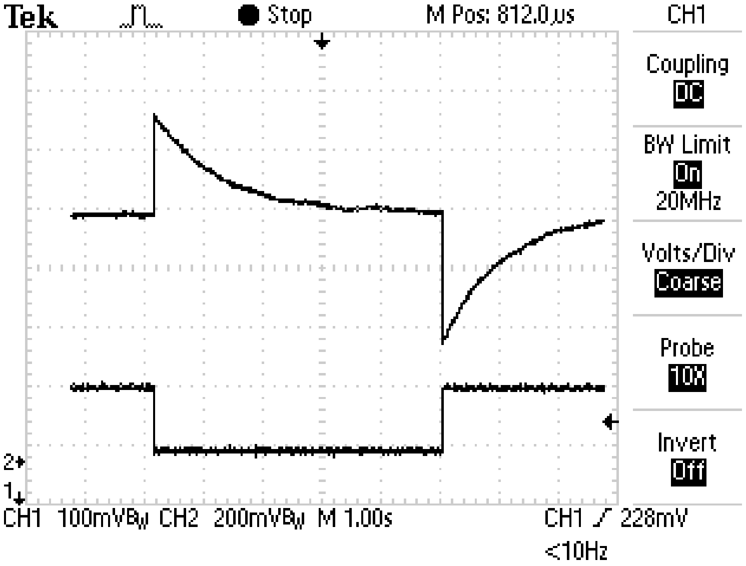


Figure 4.16: The simulated AC response with extracted parasitics matches well with the measured response. The difference in the low-pass corner frequency suggests that the estimated load capacitance (15 pF) is somewhat too low.


Tek ● Stop M Pos: 812.0µs

CH1

Coupling DC

BW Limit On 20MHz

Volts/Div Coarse

Probe 10X

Invert Off

CH1 100mVB_W CH2 200mVB_W M 1.00s CH1 ⁄ 228mV
<10Hz

Figure 4.17: A step response of the AFGA (lp). The input step size is 100mV. The output goes rapidly in the opposite direction after the edge in the input signal (inverting amplifier) and then returns to the equilibrium state at rate determined by the gate currents (droop rate).

times the closed loop ac gain of the amplifier. The floating-gate capacitance C_{tot} is charged (or discharged) and the floating-gate voltage V_{fg} reaches a new equilibrium state. The V_{fg} and the output would remain in this new steady state until the next change in the input, if all of the capacitors (including the internal capacitors of $M4$) connected to the floating node were made of perfect insulators. However, the V_{fg} returns to the initial steady state in a way (exponential decay) familiar from the natural response of an RC circuit due to the finite tunneling conductances (g_{tunnel}). Obviously, the gate current per unit area through the thick oxide devices is much smaller than that of the thin oxide device. Besides being exponentially related to the oxide thickness t_{ox}, the g_{tunnel} is also linearly related to the gate area. The gate area of the thick oxide devices has to be made larger in order to have the same capacitance value. This levels off the differences in tunneling conductances to some extent. In principle, the sum of gate currents i_{Gtot} could be calculated if the exact amount of C_{tot} was known by using the simple relation:

$$\frac{dV_{fg}}{dt} = -\frac{i_{Gtot}}{C_{tot}} \tag{4.5}$$

In this case, C_{tot} can be approximated to be 15 fF, but it is much more difficult to estimate $\frac{dV_{fg}}{dt}$ if only $\frac{dV_{out}}{dt}$ can be determined from the slope of the step response. Answering that question would require writing down differential equations including tunneling conductances in parallel on each capacitor. However, as the main goal regarding the AFGA was to gain practical proof-of-concept data, the development of analytical equations describing the cutoff frequencies is not included in this thesis.

4.4 Constant Charge Injection Implemented in 130 nm CMOS

Fig. 4.18 shows a prototype circuit featuring bidirectional programmability with IIHEI (fixed current bias through the injection transistor) and F-N tunneling. It was designed and implemented in 130 nm standard CMOS. The purpose of this prototype was to demonstrate the availability of both methods in DSM CMOS and get the initial estimations of the charge retention capability of an FGT with sufficiently thin gate oxide.

At the schematic level there are no significant differences between the presented prototype circuit and those presented in [38, 39] so that the functionality is similar. However, off-chip high voltage sources are used in this work instead of on-chip charge pumps. The operation principle is briefly revised in the following. The start-up circuit $M1$-$M3$ is to ensure the proper function of bootstrapped current source $M4$-$M8$. In read-out mode, $M17$ is configured to conducting state and $Vinj$ and $Vtun$ are set to vdd and gnd respectively. Control voltage fb is set to gnd, so that “the bottom plate” of capacitor $M13$ is grounded [3]. The current through $M16$ is determined by the floating gate voltage which is approximately: Q_{fg}/C_{M13} by assuming C_{M13} is dominating the total capacitance connected to the floating gate. In write mode, electrons are added to the floating node by IIHEI mechanism. This is done by setting fb to vdd so that “the bottom plate” of capacitor $M13$ is connected to the drain of $M11$ and $M10$ closing the feedback loop around $M10$ ($M10$ is virtually diode connected). Current through $M10$ is now set by M11 independently of Q_{fg}. Furthermore, $M10$ mirrors this fixed bias current through $M9$. A bias current of 1 μA was used in this design. Electrons are injected to the floating gate by applying a small negative voltage pulse $Vinj$ to the drain of $M9$.

Applying negative voltage pulses to the drain of injection transistor is in this case possible because of a direct connection of drain of $M9$ to an unprotected IO-pad. This further simplifies the programming scheme, because the source of $M9$ (vdd) and $Vtun$ can be kept unchanged during injection cycle (no need for ramp up nor ramp down). Alternatively, it would have been possible to use an on-chip charge pump for generating negative voltages as was done in the original reference.

Electrons are removed from the floating gate with the F-N tunneling mechanism by applying a high positive voltage $Vtun$ to the tunneling junction. Typical applied tunneling voltage $Vtun$ for this particular design was from 6 V to 8 V depending on the initial Q_{fg}. One advantage of this injection scheme over typical gate voltage controlled injection scheme is that it avoids a potential problem which may occur when tunneling is taken too far (V_{gs} of a pmos approaches 0 V or V_g is even greater than V_s). In a gate voltage controlled scheme, it is possible that the FGT cannot be reprogrammed with Hot-Electron injection any more, since the triggering of the injection mechanism requires a certain amount of source current in order to inject (see Eq. 2.5). Fixed injection current effectively avoids the occurrence of such a situation.

In the chip measurement setup, the output current was measured directly to ground via picoammeter. The main objective was to verify the functionality of the constant current injection and the secondary goal was to examine the effect of injection parameters. This requires high accuracy measurements of small off-chip currents. Measurement inaccuracy is due to

[3] Actually, there is a design bug: $M15$ should be nmos instead of pmos and consequently the inverter could be omitted. Now the capacitor $M13$ is not pulled down to ground, but remains the threshold voltage of $M15$ above the ground. However, this is not a fatal design bug since it causes only a DC offset that can be canceled through programming and thus has only minor effects on the functionality and more importantly did not prevent the evaluation of the circuit.

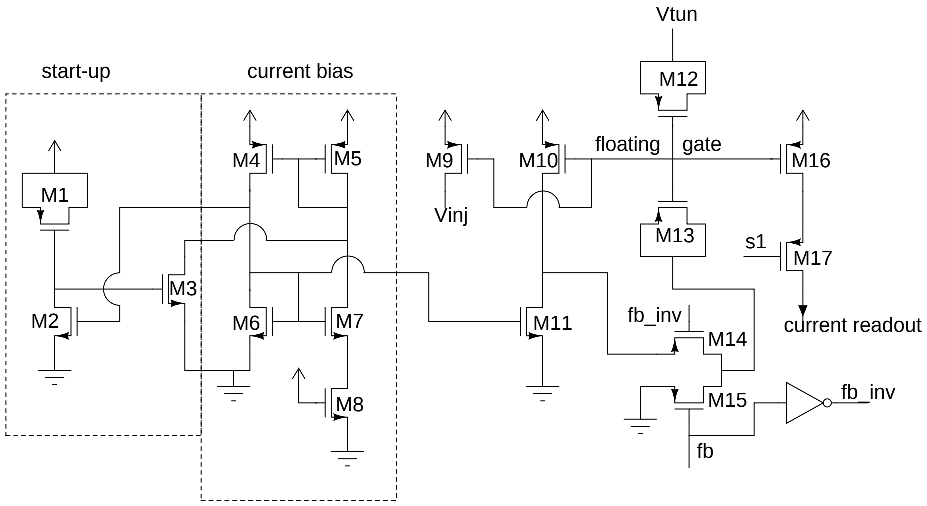


Figure 4.18: A schematic illustration of the prototype circuit featuring constant charge injection. All transistors are thick oxide IO-transistors. A sufficient current bias for injection is generated by the included current bias circuit. A separate read-out transistor $M16$ is used, thus the programming scheme is indirect. During injection phase, the feedback loop is closed and a fixed current determined by current bias circuit is mirrored through $M9$. Due to mirroring the current through $M9$ is almost independent of the stored charge on the floating gate. The injected charge is a function of bias current, the Vds voltage of $M9$ (controlled by $Vinj$) and duration of the applied $Vinj$ pulse. Injected charge can be erased by applying a high positive voltage $Vtun$ to the tunneling junction which triggers F-N tunneling. In read-out mode the feedback loop is opened and switch $M17$ is made conducting ($s1$). The programmed current flows through $M16$ and $M17$.

noise and minimum measurable current is limited by the leakage currents. In addition, the short-term drift of the output current after programming adds inaccuracy. Due to the drift, it is difficult to unambiguously determine the settling time of the output current. On the other hand, observations on the charge retention indicate a small but significant leakage current from the floating gate. For these reasons, sufficiently large steps were taken on each injection / tunneling cycle and reading of the ammeter was recorded immediately after each programming cycle, so that the changes in output current are not masked by noise, short-term drift or leakage. On the other hand, only six injection cycles per measurement were run when the influence of injection parameters was examined to minimize the effect of accumulated trapped charge (Fig. 4.20). It is important to minimize the trapped charge since it modifies the effective threshold voltage and thus reshapes the iv-curves. Trapped charge occurs during F-N tunneling when the insulator is exposed to repetitive high voltage stress. F-N tunneling is in turn required after each measurement to restore the output current to the same initial condition in order to get comparable results.

Figure 4.19 illustrates the increase in drain current of $M16$ as a function of applied consecutive injection cycles. The effective threshold voltage of p-type $M16$ is lowered as more electrons are added to the gate. Curve fitting with a second order polynomial matches the curve shown in the lin-lin scale very well. Equation 2.1 combined with the simple square-law relation of drain current to V_{th} (for a saturated transistor operating above the threshold) suggests that the change in floating gate charge is constant per injection cycle. This agrees with the observation made in [38]. Indeed, the square law relation is not accurate but satisfactory for illustrating trends. From constant change in charge per injection cycle it follows that the amount of injected charge is independent of the operation point of the transistor (initial stored charge).

Three graphs in Fig. 4.20 show the effect of injection parameters on the injection efficiency. The initial current in this measurement was chosen to be 14 μA, because the iv-curve is sufficiently linear between 10-30 μA. This way the influence of parameters can be estimated easily from the slopes of the lines. The output current was measured after each injection cycle. The top and middle graphs illustrate the influence of pulse count per injection cycle p and pulse duration pd. It can be readily seen that the dependence of these parameters on accumulated charge is linear as can be expected. According to the Eq. 2.5, the V_{sd} is exponentially related to the gate current which can be seen from the bottom graph. An alternative representation is provided in Fig. 4.21, where the final values of $Iout$ after six injection cycles are plotted as function of these same three parameters.

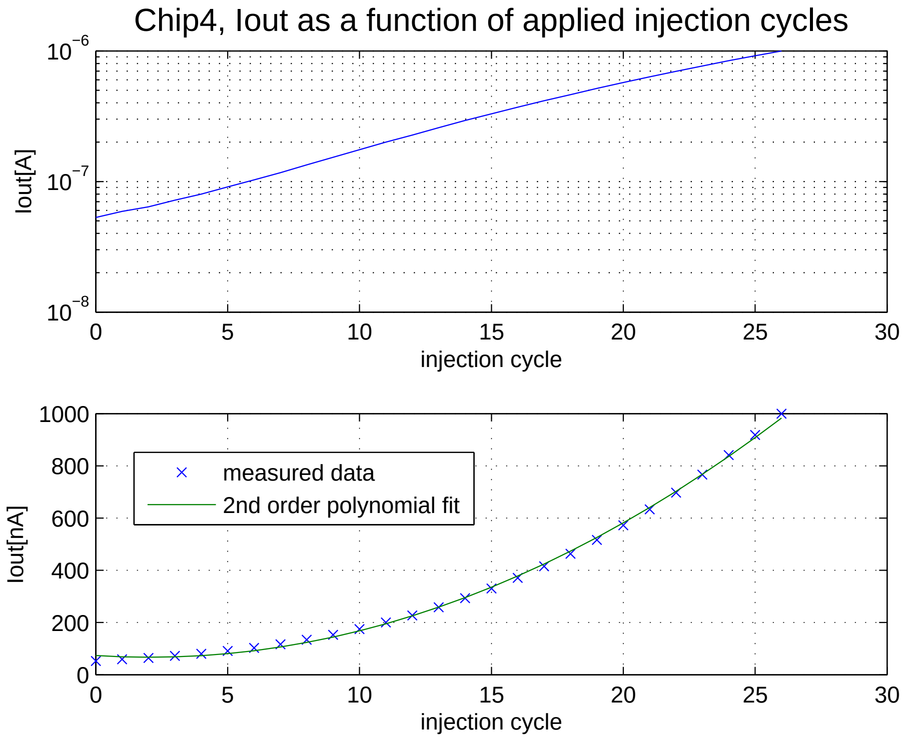


Figure 4.19: The increase in drain current of $M16$ as a function of applied injection cycles with the following parameters: Vdd=2.52 V, $Vinj$=-0.98 V, p=100, pd=550 μs, T=1100 μs. where p is the number of applied injection pulses per injection cycle, pd pulse duration and T pulse period. The initial output current after tunneling is approximately 50 nA. The output current plotted on a semi-log scale shows a linear increase and ends at 1 μA after 27 injection cycles (upper curve). The same curve plotted on a linear scale with a second order polynomial fit is shown in the lower figure.

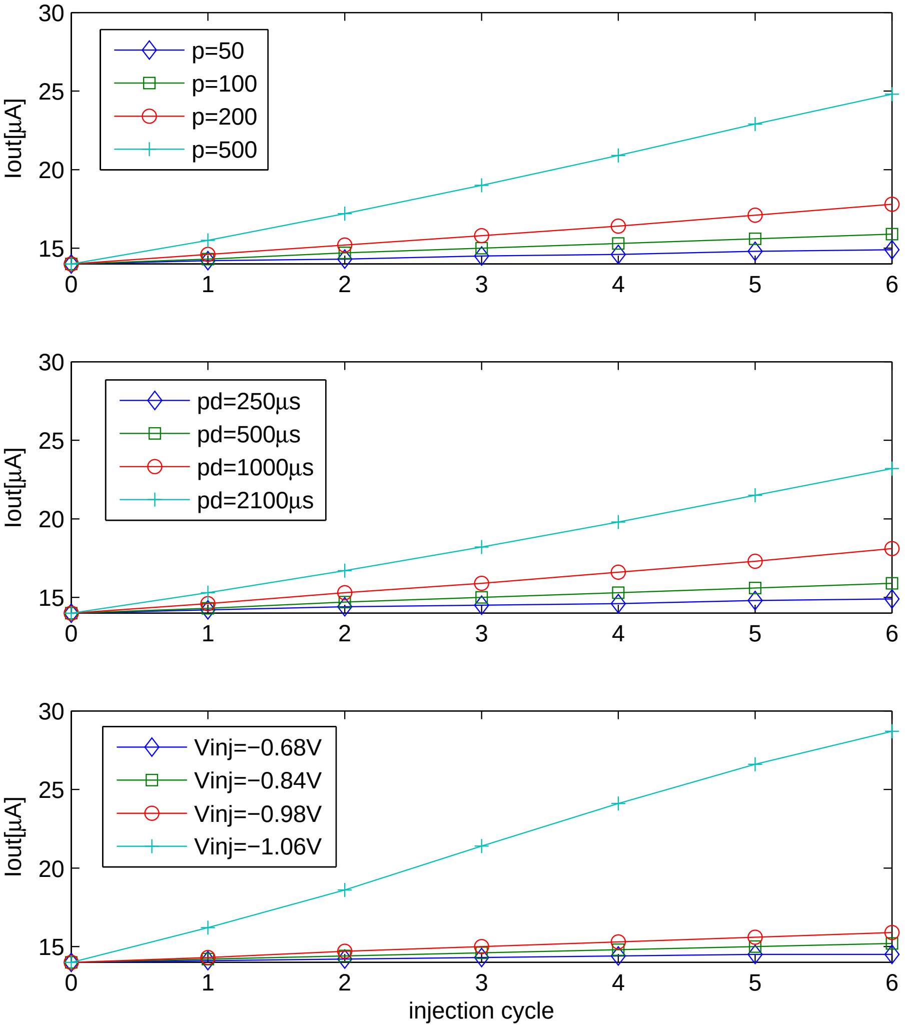


Figure 4.20: The increase in drain current of $M16$ (see Fig. 4.18) as a function of applied consecutive injection cycles when the injection parameters are varied. For the number of applied injection pulses per injection cycle p and pulse duration pd the relation is linear and exponential for the drain-source voltage V_{sd} (V_{inj}), respectively.

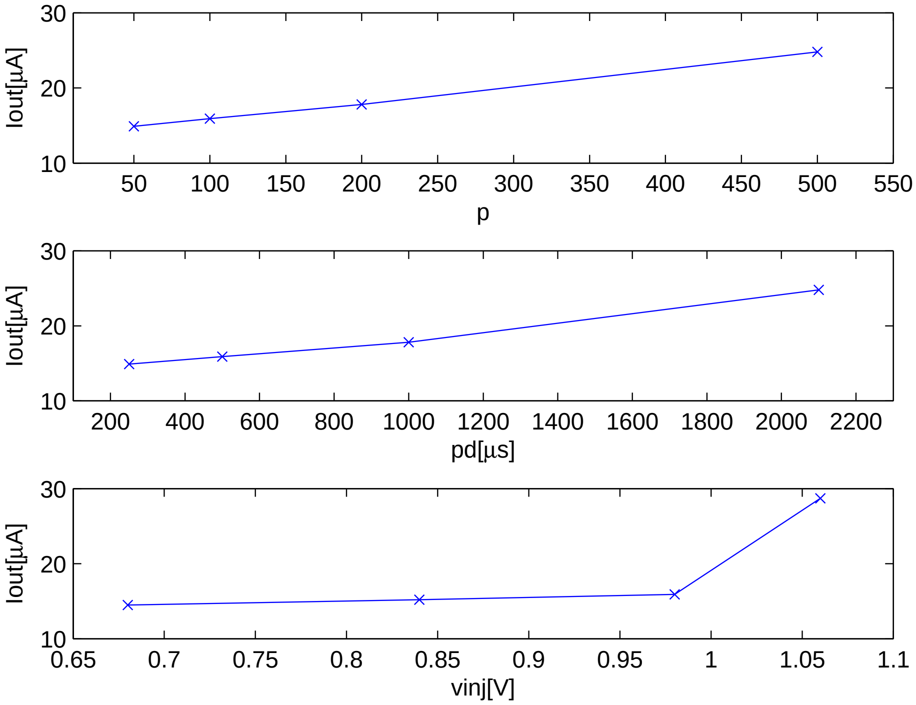


Figure 4.21: $Iout$ after six injection cycles as a function of pulses per injection cycle p, pulse duration pd and injection voltage V_{inj}, respectively.

Chapter 5

Introduction to Spiking Neural Networks

This chapter provides a short introduction to biological and artificial neural networks and the essential concepts related to this field. It is not meant to be exhaustive, but is included to provide relevant background information and references for those not familiar with the neural networks and neurobiology.

5.1 Background

The structure and function of the brain have been studied intensively over the past hundred years in order to solve the mystery of how information is processed in the brain. Despite the huge amount of accumulated data, the way the brain works is not understood that well yet. However, these pieces of knowledge from neuroscience have been used as inspiration for new techniques and algorithms in Artificial Neural Networks (ANN). ANNs were not intended to perfectly model biological neurons, but rather to aim for computational effectiveness to perform complex non-linear computations. Recently, part of the research has shifted toward creating biologically realistic models in order to examine and understand the information processing in the brain. For example, the Blue Brain project attempts to reverse engineer the mammalian brain [53]. By mimicking the way that brains compute, researchers aim at constructing computation platforms with comparable power efficiency, learning and adaptation ability to the brain.

There are considerable differences between the biological and silicon based computation mediums and thus the biological models cannot be just duplicated in silicon. However, as highlighted in [54] there are also many similarities between the biological channels and semiconductor channels of the transistors which encourages the implementation of neuromorphic analog hardware. For example, the biological channel controls the flow of ions across the cell membrane, while the semiconducting channel beneath the gate of a transistor controls the flow of charge carriers. The birth of *neuromorphic engineering* and *physics of computation* is centered around the pioneering collaboration of the trio Carver Mead, John Hopfield and Richard Feynman, and their efforts to study how animal brains compute. Carver created the first neurally inspired chips, including the silicon retina and chips that learn from experience [57]. Today, there are several research groups around the world making an effort to achieve brain-like computing in silicon. For example Brains in Silicon group in Stanford University [55] and BrainScaleS [56] which is a collaboration of 18 research groups from 10 European countries. In Stanford, the idea is to design an affordable supercomputer by utilizing existing knowledge of brain function. This supercomputer serves as a tool to investigate brain function by creating a feedback loop around the fundamental, biological understanding of how the brain works. BrainScaleS approaches this problem from the opposite viewpoint by trying to extract generic theoretical principles from in-vivo biological experimentation and computational analysis to enable an artificial synthesis of cortical-like cognitive skills.

5.1.1 Human brains

The following information is collected from the book "Dynamical Systems in Neuroscience" [59]. These facts provide a good perspective of the complexity of the human brain, and help to understand the difficulties scientists have to overcome as they pursue to reveal the secrets of the brain.

1. There are about 10^{11} neurons in the human brain and a typical neuron receives inputs from over 10 000 other neurons.

2. In brains, the information-processing depends on the electrophysiological[1] properties of neurons as well as their dynamical properties. Even if two neurons in the same region of the nervous system possess similar electrophysiological features, they may respond to the same synaptic input in very different manners because of each cell's bifurcation dynamics[2].

3. The type of bifurcation determines the most fundamental computational properties of neurons, such as the class of excitability, the existence or nonexistence of a threshold, all-or-none spikes, subthreshold oscillations, ability to generate postinhibitory rebound spikes, the bistability of resting and spiking states, whether the neuron is an integrator or a resonator, and so on.

[1] Electrophysiological means the electrical properties of biological cells and tissues.

[2] Neurons are excitable because they are near a transition, called bifurcation, from resting to sustained spiking activity.

4. There are millions of different electrophysiological mechanisms of spike generation. However, there are only four different types of bifurcation of equilibrium that a system can undergo without any additional constraints. These are called saddle-node, saddle-node on invariant circle, subcritical Andronov-Hopf and supercritical Andronov-Hopf bifurcation.

The theoretical details, such as the difference between bifurcations, are unimportant from the perspective of this thesis and therefore are not explained further. The elementary building blocks of biological neuronal systems and their artificial counterparts are discussed in the following.

5.1.2 Neuron

One essential concept in brain science is the concept of a neuron. A neuron can have either inhibitory or excitatory outputs, but cannot have both (Dale's law) [62]. Cortical neurons of the brain are of special interest because all the higher-level psycho-physical functions, such as sensory perception, object- and event-representation, planning, and decision making are believed to take place in the neocortex [63].

An artificial neuron is an abstraction of biological neurons and is the basic unit in an ANN. There are many mathematical models with varying complexity and accuracy (or level of realism) of biological neurons. In general, the silicon neuron must be compact (small area) and efficient (low power) to allow as many of them as possible to be put on a single silicon chip. Abstractions of neuronal models can be divided into three categories: 1) rate-coding model 2) spiking model and 3) compartmental model, each having their own characteristics. A rate-coding neuron model is the most inaccurate but on the other hand the most tractable at the same time, while compartmental models represent the other end. Spiking models are a kind of compromise between these two extremes and are equipped with dynamical properties.

One of the earliest spiking neuron models proposed is the Integrate-and-Fire (I-F) neuron [64]. Despite its age, it, or its variant leaky I-F [65], is still widely used and extremely useful especially in large scale experiments due to its simplicity and compact size. Models of spiking neurons can be ranked according to the number of neuro-computational features and implementation efficiency. By adopting these two criteria, the most efficient implementation with least number of features is the I-F model while Hodgkin-Huxley (H-H) [66] model is in the other end. An example of a more recent development in neuron modeling is the soliton model [67], which is based on thermodynamic theory of nerve pulse propagation in which the action potential is a reversible electromechanical soliton [3]. In [60], 20 of the most prominent neuro-computational features existing in biological spiking neurons in neocortex are reviewed. An analog implementation of the neocortical spiking neuron which captures all of these 20 features with tunable parameters using only 14 transistors is presented in [69]. The modified version of this circuit is adopted in this thesis and presented in Chapter 6.

The I-F and H-H spiking neuron models are presented in Fig. 5.1 and their operation principles are briefly following. When the I-F neuron of Fig. 5.1 a) is stimulated with current pulses, the membrane voltage $Vmemb$ increases with time until it reaches a constant threshold Vth. When the threshold is exceeded, a delta function spike occurs and the $Vmemb$ is reset to its resting potential, after which the model continues to run. The firing frequency of the model is a linear function of the input current pulse rate and amplitude. The leaky model includes a shunt resistor *Rleak* which constantly discharges the *Cmemb* and thus adds one dynamical dimension to the model. The electrical schematic for H-H model is presented in Fig. 5.1 b). Hodgkin and Huxley discovered that the total current in a biological neuron is a sum of currents in sodium (Na+) and potassium (K-) ion channels and they both have separate dynamics and opposite directions. The inward current carried by Na+ ions is fast, while the outward current carried by K- ions is activated more slowly. The dynamics of the variable resistors in their model is hard to implement with hardware because the conductance is a function of $Vmemb$.

5.1.3 Synapse

In biological as well as artificial neural networks, the neurons are interconnected through weighting elements, called synapses. The biological synapse was first thought to only just transfer a signal from axon (presynaptic “wire”) to the dendrite (postsynaptic “wire”), but it has proved to perform more complicated signal pre-processing tasks and has a crucial role in learning and adaptation [58]. It has been discovered that the long- and short-term history of the synaptic activity influences the role of a synapse as a pre-processor.

An artificial abstraction of the biological synapse is an element which is capable of weighting the incoming signal by an internally stored value and adapting this weight based upon a particular learning rule. The simplest weighting used in ANNs is binary, i.e., there is a connection between two neurons or there is not. A continuously valued weighting scheme allows more sophisticated learning algorithms to be implemented and is relatively easy to implement with analog hardware. Dozens of different learning rules

[3] signals which travel along the cell’s membrane in the form of certain kinds of sound pulses are called solitons.

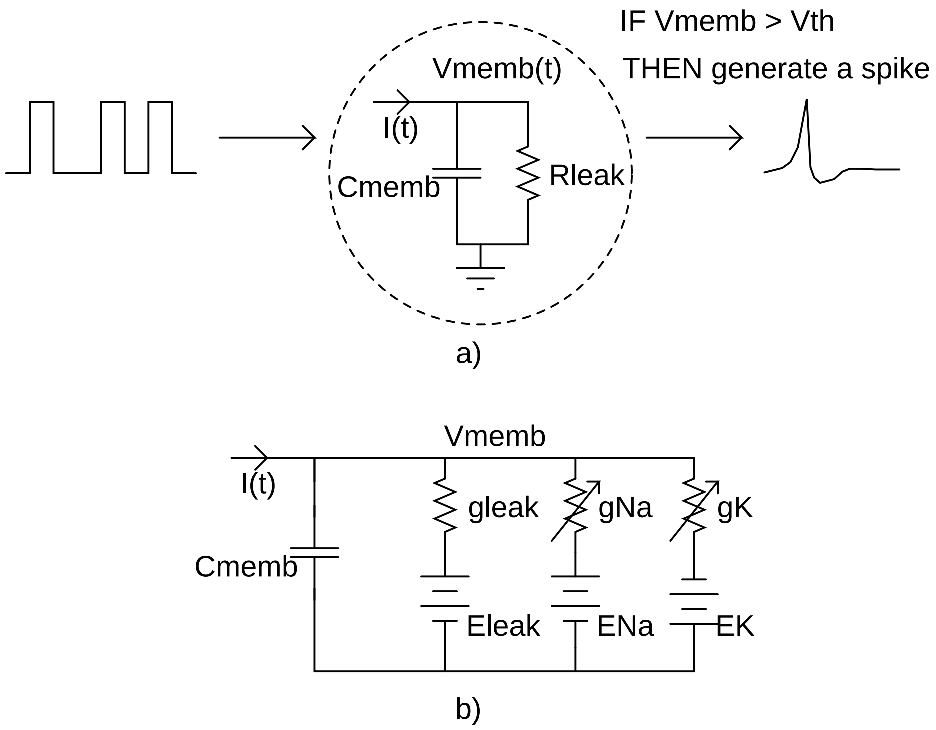


Figure 5.1: a) the I-F neuron model (only a leaky I-F model includes *Rleak*). b) the H-H neuron models the ionic sodium and potassium channels.

have been proposed, typically as functions of correlations of signals passing through each synapse. Due to the role in learning and adaptation, synapses can be regarded as key components in neural networks. The requirement for small size and low power is pronounced with synapses, simply because synapses outnumber neurons in a typical neural network.

5.1.4 Spiking Neural Network

One can distinguish three generations in the evolution of artificial neural networks. The first generation neurons communicate by sending a binary high signal if the sum of its weighted incoming signals rises above a threshold value. The second generation neurons are upgraded so that they use a continuous activation function. The computation in the first and second generation ANNs is based on the so-called rate coding, where a higher rate of firing correlates with a higher output signal (integrating-type). A spiking neural network is a third generation ANN that abandons the rate coding and instead uses individual spikes in computation. An extension to the level where every spike counts opens up a whole new range of information coding options as well as introduces new challenges. An excellent introduction to SNN is given in [58].

In SNN neurons fire spikes when they reach a certain threshold. These spikes travel to other neurons which, in turn, increase or decrease their potentials in accordance with this signal, thereby approaching or drawing away from the threshold level. This model of a neuron as an integrator with a threshold does not apply to all biological neurons, but is a reasonable compromise between complexity and biological relevance. Furthermore, the biological neurons have a variable threshold that depends on the prior activity of the neuron while artificial neurons have typically a fixed threshold.

In the rate-coding, the key parameter is the mean firing rate of neurons. In the case of SNNs, the temporal structure of the spike train is examined. For example, the time intervals in which the neurons will generate their first spike for a given input stimuli can be used for coding data. One widely used coding scheme that has come in handy with SNNs is rank-order coding. Rank-order coding is an abstraction of true temporal coding, because it is not concerned with the exact timing of the spikes but only with their relative order of firing in a layer [62]. A protocol called Address-Event-Representation (AER) implements a rank-order coding scheme and was used for example in the CAVIAR project to interface neuromorphic hardware [70].

To date, one of the largest neuromorphic chips is implemented within the CAVIAR project [61] and includes 45k neurons and 5M synapses (actually four chips connected together). The gap expressed in number of neurons is enormous when compared with human brains. Also, the number of neurons (or synapses) between different implementations are not directly comparable because of the tradeoff between the neuron complexity and the accuracy of the model. A more detailed neuron description (increased biological relevance) usually leads to large and inefficient circuitry, thus allowing only an implementation of a very simple network while a simple neuron description fails to capture the dynamics of biological neurons. Another issue arises from the differences in each cell's bifurcation dynamics. In the literature search only implementations of artificial neuronal systems which consist of neurons with homogeneous bifurcation dynamics were found while biology suggests systems that have heterogeneous dynamics.

The CAVIAR is not the only project where large scale neuromorphic chips or systems have been developed. Noteworthy results, with slightly different approaches, have also been achieved in the following projects: spiNNaker [71] and FaCETS [72] (the continuation of the FACETS project is called BrainScaleS). In the former project, the basic building block is a multicore processor known as the "spiNNaker chip". It comprises 20 processing cores where each core is a complete sub-system. The cores are connected together via a Network-on-Chip (NoC). In addition, the chips can be linked to each other so that the system can be extended to include thousands of cores. In the latter project, the idea is to connect neurons on a wafer level. That is, the interconnections between the individual reticles on a wafer are formed by depositing an additional metal layer on top of the wafer in a post-processing step. After the post-processing step, a single wafer contains $4 \cdot 10^7$ synapses interconnecting up to 180 000 neurons.

5.1.5 STDP

A specific learning rule is needed for the synaptic weight modification. Hebbian learning is a well known mechanism that is based on the correlated activity of neurons [73]. The weights should change relatively slowly in order to achieve meaningful and stable operation. The term *long-term synaptic plasticity* is often used to express this characteristic. One mechanism that has been experimentally observed in biological synapses and exhibits long-term synaptic plasticity is Spike Timing Dependent Plasticity (STDP) [74].

Many of the learning methods developed for use with STDP models are based on Hebbian learning with an additional temporal aspect. Thus, the weights are modified according to pre- and postsynaptic spikes: if the presynaptic spike comes before the postsynaptic spike, the weight is increased; otherwise it is decreased (Hebbian correlation). The modification of the weight is stronger when the delay between pre- and postsynaptic spikes is small (temporal aspect). The temporal aspect of the STDP mechanism creates a built-in competition mechanism that stabilizes the neuronal system and thus avoids the problem with the pure Hebbian learning endlessly strengthening effective and weakening ineffective synapses (positive-feedback process)[58].

Chapter 6

FGT Implementation of SNN

A floating-gate-based implementation of a synapse within the context of Spiking Neural Network (SNN) is presented in this chapter.

6.1 Floating-gate Transistor Based SNN

6.1.1 Motivation

The motivation to implement an SNN with analog hardware was to investigate the realization of a STDP rule for synaptic connections for three reasons: 1) STDP outperforms the traditional Hebbian synaptic plasticity [75], 2) the neuromorphic paradigm is attractive for future nanoscale computation due to massive parallelism, and 3) potentially many robustness issues can be side stepped thanks to the adaptation and learning functionality of the network, which makes it more failure-tolerant [76]. Emerging nanodevices may play important role in future artificial neural networks. The nanoscale synapses and wires enable significant improvement in the efficiency of connectivity. However, many of these technologies are still in their infancy and not widely available for researchers. However, currently available DSM FGT devices can be used to effectively realize a new biphasic STDP learning mechanism.

The key observation is that when FGT synapses are programmed with F-N tunneling, the synapse appears as a two-terminal device with characteristics similar to a memristor [6]. A memristor is currently one of the most promising nanodevices (see Chapter 7). Thus, the proposed learning method is also applicable to networks comprising memristors for synaptic connections. Besides the work of the author [78], the use of biphasic learning pulses to implement STDP rule in neuromorphic hardware was proposed independently also by two other groups [79, 80]. Three different synapses all employing biphasic learning pulses for realizing STDP are shown in Fig. 6.1.

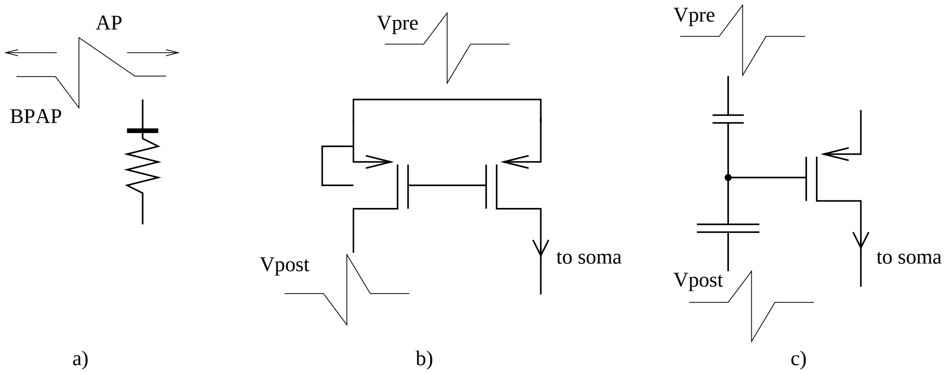


Figure 6.1: a) memristor synapse proposed in [80]. Backwards traveling Back Propagation Action Potential (BPAP) sets the programming ON/OFF, forward traveling Action Potential (AP) sets the weight of the synapse. b) A FGT synapse proposed in [79] employs an indirect programming scheme and combination of IIHEI and F-N tunneling. c) A synapse proposed in this work has separate signal and programming lines and uses F-N tunneling for weight modifications.

The study in [81] presents a CMOS implementation of a biphasic pulse generator. However, the peak-to-peak amplitude is only from -0.9V to +0.9V, which is too low for programming FGTs as well as memristors with current technology. On the other hand, the approach presented in [79] is well suited for the weight modification of FGT synapses but is improper for memristor synapses due to mismatch of methods used for weakening and strengthening the weights (combination of electron injection and electron tunneling). However, it is not claimed that the method proposed in this work (based on simultaneous post- and presynaptic biphasic pulses) to modify synaptic connections by STDP rule is exactly how biological action potentials affect on synapses, but successfully approximates similar functionality by taking into account the characteristics of the implementation medium.

In recently developed hybrid CMOS / Nano circuits, such as CMOL and FPNI, which will be reviewed in Chapter 7, nanodevices are arranged into a single layer as an add-on to a CMOS sub-system. This arrangement allows very dense architectures and enables a great increase in the number of synapses that can be implemented on a single chip. With FGTs implemented in a bulk CMOS, this is not feasible, since FGTs are CMOS devices and thus can not be placed above other CMOS devices. A 3D process would effectively allow this.

Architectures with FGT-based synapses share many design challenges with memristor-based synapses, for example they both need high voltages for weight modification. Furthermore, typically the implementations of FGT synapses, such as [36], have been asymmetric: the input signal is multiplied with the weight stored on the gate producing an output current at the source/drain, thus providing isolation between presynaptic and postsynaptic neurons. This isolation property prevents signals traveling back to other presynaptic neurons. Memristor-based synapses and FGT synapses with the drain and source as input an output terminals are symmetric devices which are incapable of providing isolation between presynaptic and postsynaptic neurons and thus are more like resistive diffusion networks [83]. Consequently, neural networks consisting of symmetric synapses are fundamentally different from those with asymmetric synaptic connections. The similarities and differences of memristors and FGTs will be revisited in more detail in Chapter 7.

On this account, an FGT-based implementation can be used as a demonstration vehicle that allows experiments with currently available technology. This approach helps to reveal possible obstacles and challenges that are likely to emerge with hybrid CMOS/Nano neuromorphic circuits, and to develop solutions to these problems.

6.1.2 Implementation

It should be noted that the circuit realization of the SNN presented in this thesis does not really implement the algorithm for the STDP rule but only the mechanism required to implement it. That is, the learning and communication mode are separated and thus the weights can be modified only by controlling the network with external stimuli in the learning mode: the proposed FGT synapse has four terminals but only two of them are used simultaneously making the FGT virtually a two-terminal device. Two capacitive terminals are used for weight modification in the learning mode and the drain and source of the synapse transistor are used in the communication mode to form a voltage controlled current source. This is necessary because, unlike in case of memristors, the terminals for communication and learning can not be shared. This arrangement also allows operation in different timescales on different operation modes. Weight modification requires a slower timescale but in communication mode the bandwidth can be increased by shortening the spike duration and interval. Weight modification is local and fully parallel and takes the temporal aspect into account as required by the STDP rule. During the communication mode the synaptic weights are stored as charge to the floating gates of the synapses.

The top level schematic of the prototype circuit is shown in Fig. 6.2. Digital parts of the design work with a 1.2 V supply voltage and the neural network (and ring oscillator) with 4.5 V *vdd* and 2 V *vss*. The 2 V shift in supply voltages is needed for the selected weight modification scheme which will be examined later in more detail. *vdd*2*v*5 is used only in voltage level shifters as an intermediate supply to facilitate conversions between different power supply domains. All input pins marked with an asterisk are wired through custom made unprotected IO-pads, because suitable IO-cells were not available in IO-libraries provided by the foundry. The chip communicates with a XEM3001 FPGA integration module (Opal Kelly) that collects the spike data and controls the chip. The FPGA module is hosted by a standard pc through the usb-port. Necessary supply / off chip bias voltages are generated by standard voltage regulators on the PCB.

The synapse and the neuron with a built-in charge pump are the most essential building blocks in the prototype SNN chip. The neural array is a fully connected (256 FGT synapses) two layer network with 16 neurons in both layers (one neuron in the input layer has only inhibitive outputs). It has a digital AER interface and decoders for addressing individual neurons. Additional features are internally generated bias voltages, per neuron charge pumps for generating tunneling voltages, a ring oscillator for charge pump clocking and voltage level shifters between different power supply domains. Due to IO-pad limitations, only spike events (*NOR_out*) with associated addresses ($out0 - out3$) and high voltage output (*hv_out*) of a test neuron can be observed.

The implemented chip was a multi-project IC which set additional challenges and design constraints in terms of area and IO pin count. For example, due to included RF circuits, the chip is packaged in a Low profile Fine pitch Ball Grid Array (LFBGA). In practice, for a relatively complex system, having only 22 IO pins available limited the scaling of the SNN array more than area constraints. Limited IO adds some complexity to the design but more importantly severely restricts the debugging of the circuit because internal nodes can not be accessed (except the high voltage output of the separate test neuron).

Neuron

The presented neuron in Fig. 6.3 is partly based on the artificial neuron presented in [69]. The illustrated schematic is divided into three blocks for the easier identification of different functionalities of the neuron circuit. The "integrator and spike generator" (1) block is a simplified version of that presented in [69]. The simplified version discards most of the 20 neuro-computational features of the original design and thus reduces to the basic I-F model. This is due to two reasons: 1) the emphasis was on implementing the STDP feature for synaptic connections using biphasic learning pulses for which simple I-F dynamics is sufficient and 2) it is still unknown how information processing should be organized with heterogeneous spiking dynamics. Thus, the parameter space can be decreased by omitting unneces-

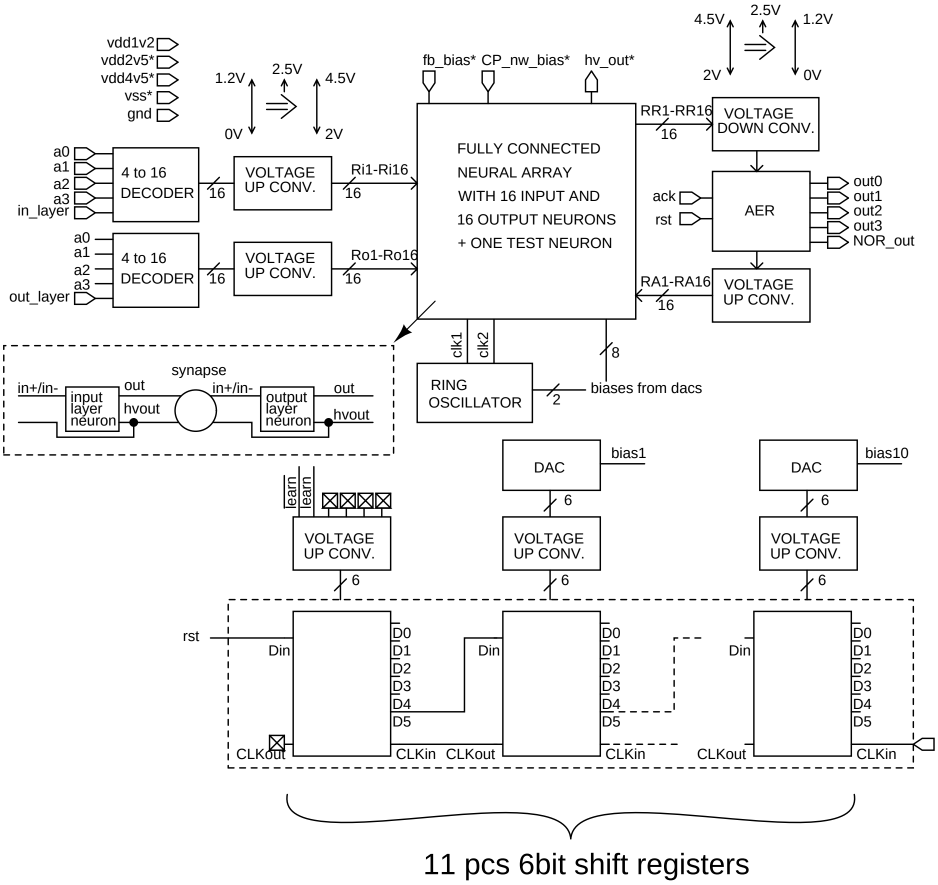


Figure 6.2: A top level schematic of the implemented SNN. It features a fully connected (256 FGT synapses) two layer network with 16 neurons in both layers (one neuron in the input layer has only inhibitive outputs), a digital AER interface, decoders for addressing individual neurons, internally generated bias voltages, per neuron charge pumps for generating tunneling voltages, a ring oscillator for charge pump clocking and voltage level shifters between different power supply domains. Due to IO-pad limitations, only spike events (*NOR_out*) with associated addresses ($out0 - out3$) and high voltage output (*hv_out*) of a test neuron can be observed.

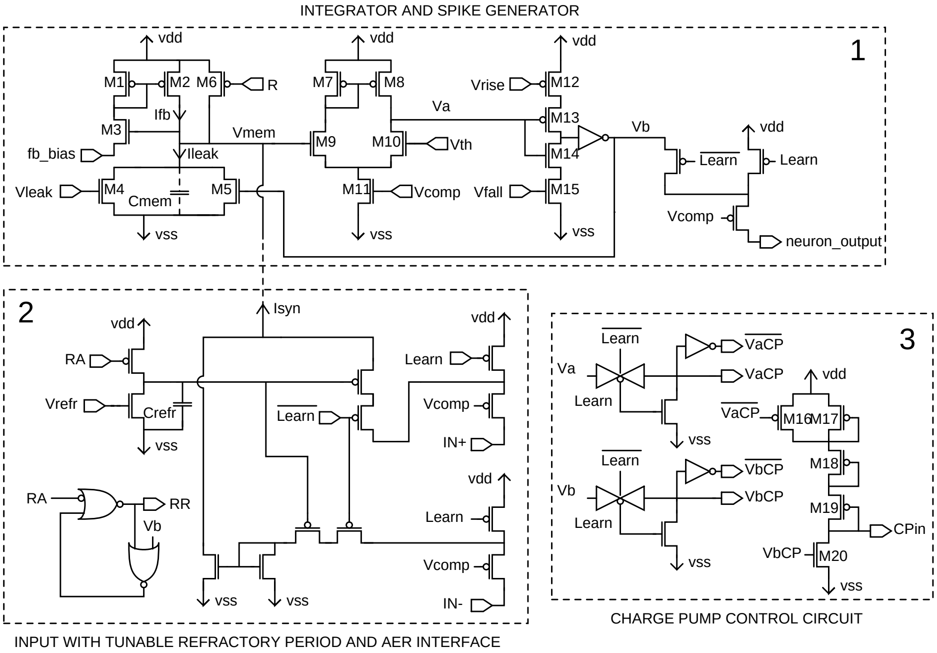


Figure 6.3: A schematic illustration of a leaky I-F neuron with AER interface. FGT synapses (not shown in the figure) form resistive paths between the presynaptic neuron (*neuron_output*) and postsynaptic neurons ($IN+$ for the excitatory or $IN-$ for the inhibitory connection). The strength of the synaptic connection between two neurons is determined by the floating-gate charge that can be modified during the learning mode by applying the high voltage biphasic pulses generated by the charge pump. The circuitry for controlling the charge pump is visible as well (every neuron has a built-in charge pump). The circuit operates with supply voltages of 4.5 V (vdd) and 2 V (vss).

sary features from the point of view of this research. An interesting remark is that it was very easy to make the original neuron circuit, designed in 0.35 μm CMOS ([69]), to work in the 65 nm technology — a direct manifestation of the scalability of neuromorphic analog hardware over different technology nodes.

Communication Mode

The operation of the neuron in communication mode is the following. The *Learn* signal is first set to 1. The membrane capacitor $Cmem$ integrates the sum of excitatory $IN+$ and inhibitory $IN-$ postsynaptic input currents and generates a spike when the membrane potential $Vmem$ exceeds the spike generation threshold voltage set by the source voltage fb_bias of $M3$. When the threshold is exceeded the membrane potential increases rapidly due to the positive feedback loop formed by the current mirror ($M1$ and $M2$), which causes the comparator ($M7$-$M11$) to flip. Vth sets the turnaround point of the comparator (spike detection threshold) and $Vcomp$ controls the bias current of the comparator. $Vleak$ at the gate of $M4$ sets the leakage rate of the membrane potential. $Vmem$ can be derived from the following equation:

$$Cmem\frac{\mathrm{d}Vmem}{\mathrm{d}t} = Isyn + Ifb - Ileak \tag{6.1}$$

where $Isyn$ denotes the sum of postsynaptic currents, Ifb the current flowing through the current mirror and $Ileak$ the leakage current through $M4$. After the spike event, $M5$ resets the neuron by discharging $Cmem$.

The size of $Cmem$ can be in the order of femtofarads if the synapses operate in the subthreshold region allowing only nanoampere currents to be delivered to postsynaptic neurons. In this particular design there is no explicitly implemented $Cmem$ device, it consists of parasitic capacitances only. The current-starved inverter ($M12$-$M15$) is used to control the lengths of voltage pulses Va and Vb (Va is meaningful only in the learning mode).

In communication mode, $Vfall$ is set to vdd and $Vrise$ controls the length of the emitted spike (Vb) and the duration of the reset period caused by $M5$. The generated spike is delivered to synapses connected to the node denoted as $neuron\_output$. The pmos cascode transistors sharing the bias $Vcomp$ at the input and output of the neuron are not absolutely required but improve the isolation between pre- and postsynaptic neurons.

The distribution of neuron inputs to excitatory $IN+$ and inhibitory $IN-$ is fixed, thus the connectivity of the network must be determined in the design phase. In this case, one of the neurons in the input layer is connected to the inhibitory input of neurons in the output layer and the rest of the connections are excitatory. The fixed connectivity of the network can be considered as an acceptable compromise in the prototype chip, because it allows the testing of the concept without additional complexity required for multiplexing the excitatory and inhibitory connections for each neuron.

The rest of the presented neural circuit is for implementing the AER interface controlling the charge pump during the learning mode. A spike can be induced also from outside the network by activating the R signal. A spike event leads to a high AER row request signal RR. RR is latched until the acknowledgement signal RA (active low) resets it. The idea for the realization of the refractory period is adapted from [84]: When RA is activated (low) for a short period of time $Crefr$ is charged. $Vrefr$ sets the discharge rate of $Crefr$ when RA is set back to a high value and defines the length of the refractory period. During the refractory period no current can flow into the neuron. The implementation of the neuron can lend itself

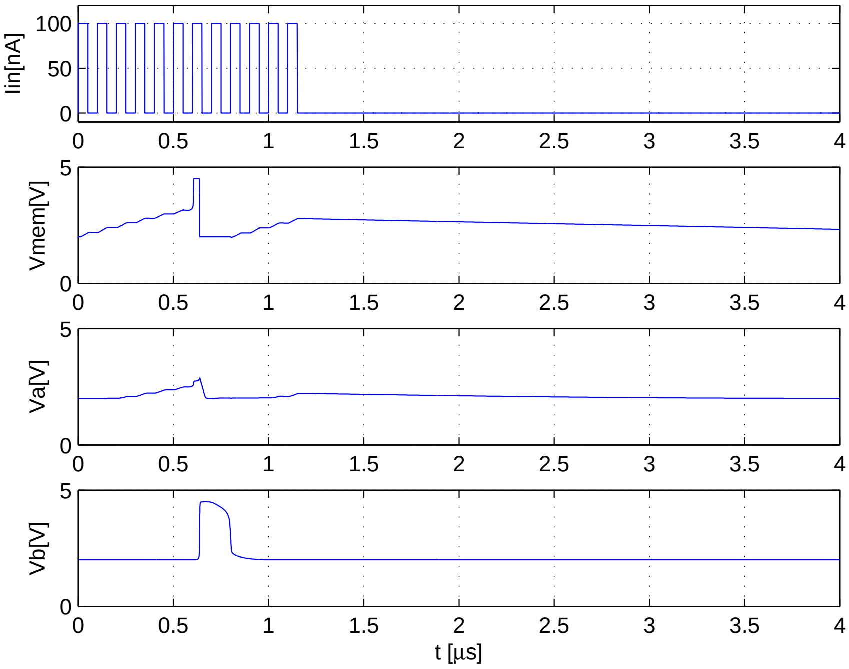


Figure 6.4: A transient response of a neuronal circuit (Fig. 6.3) for a 100 nA current stimulus applied to $IN+$ when configured to the communication mode. $Cmem$ integrates the current until $Vmem$ exceeds the predefined threshold voltage and generates a spike Vb. Va is meaningful only in the learning mode. If the stimulus is turned off before the spike event, $Vmem$ decreases at a rate determined by the $Vleak$.

to larger arrays by simply duplicating the AER switches R and RA and making small modifications to the realization of the circuit that generates the RR signal so that row and column addressing is allowed.

The simulated example waveforms of selected signals are shown in Fig. 6.4 for the neuron circuit presented in Fig. 6.3 configured into communication mode. The neuron is stimulated with a 10 MHz current stimulus Iin (equivalent to $Isyn$) consisting of 100 nA pulse train. The neuron fires a spike Vb when $Vmem$ exceeds the threshold voltage. The small refractory period ($\sim$ 200 ns) prohibits the integration of the first two current pulses after firing a spike. When the stimulus goes off, the $Vmem$ slowly returns to the resting potential due to leakage current $Ileak$ through $M4$.

Learning Mode

By setting the $Learn$ signal to 0 the neuron is configured into learning mode which allows the modification of synaptic weights. By doing so, the drain and source terminals of synapse transistors are pulled to vdd to prevent unwanted tunneling from taking place and transmission gates deliver the control signals Va and Vb for the charge pump. No current can flow into the neuron and spikes can be induced only externally by activating the R signal. Now the current starved inverter is used to slow down the timescale so that the charge pump has enough time to generate a sufficiently high voltage for F-N tunneling during the time Va is high. This is done by setting the bias voltage $Vfall$ so that $M15$ operates in the subthreshold regime. $Vrise$ controls the length of the Vb pulse similarly in the communication mode.

In the learning mode, the length of the Vb pulse sets the reset period of the charge pump. The circuit consisting of transistors $M16$-$M20$ is responsible for slowing down the output of the charge pump when the output returns to the DC-level after reset so that the rise times from DC-level to maximum value and from ground to DC-level are balanced. The speed regulation is based on adapting the input voltage $CPin$ of the charge pump according to the control voltages Va and Vb. The functionality of the charge pump is explained in more detail in Section 6.1.2. Simulated example waveforms of selected signals are shown in Fig. 6.5. It can be seen that the length of Vb pulse is not affected but the length of Va is increased to about 10 μs. Also shown are the changes in $CPin$ during the learning period. A sequential activation of R signals for a selected pair of neurons, one in the input and one in the output layer, effectively triggers the STDP mechanism. The per neuron charge pumps generate a biphasic pulse pair which are superimposed at the synapse. The strength of the weight adaptation is adjusted by the activation delay of presynaptic and postsynaptic R signals.

The layout (see Fig. 6.6) of the neuron circuit with associated charge pump measures 30 by 60 microns. About half of the area is occupied by the charge pump that is included in every neuron. The three blocks in Fig. 6.3 are labeled with corresponding numbers 1-3 in the layout.

Synapse

The synaptic weights were implemented with FGTs and updated with F-N tunneling. The proposed synapse is presented in Fig. 6.7 and the corresponding layout in Fig. 6.8. When the network is configured to the communication mode, the voltages applied to capacitive terminals are kept constant. When neuron A emits a spike, all synapses connected to the axonal output of neuron A convert this voltage spike to a current pulse that flows to the denritic input ($IN+$ or $IN-$) of a neuron in the output layer (e.g.

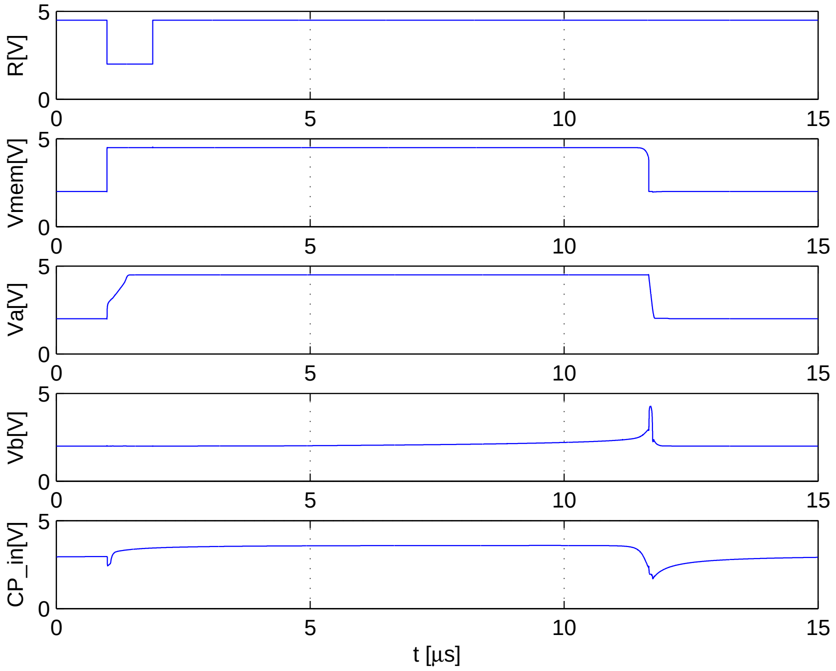


Figure 6.5: Transient response of a neuronal circuit (Fig. 6.3) for an externally induced spike event when configured to the learning mode. The neuron is made active by activating the R signal. Signals Va and Vb control the charge pump and their lengths are adjusted by the current starved inverter. The input voltage of the charge pump $CPin$ is intentionally not constant during the learning phase.

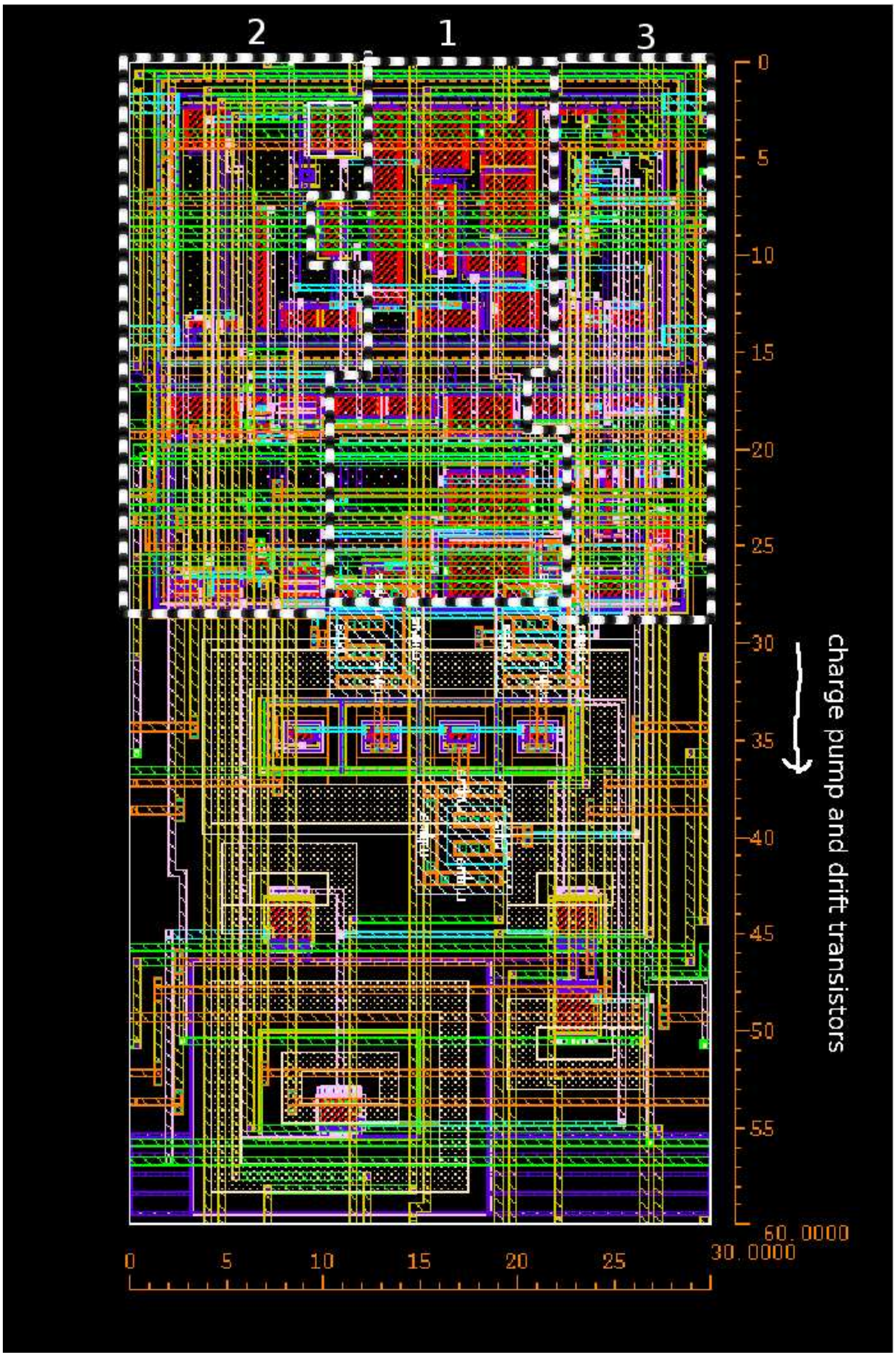


Figure 6.6: Layout of the neural circuit with a built-in charge pump. The charge pump and high voltage tolerant drift transistors occupy half of the neural circuit. The labels 1-3 refer to respective circuit blocks in Fig. 6.3.

neuron B). The amplitude of the current pulse is determined by the channel conductance of the pmos synapse transistor Mr which is set by the floating-gate voltage Vfg. In the learning mode, the biphasic high voltage pulses generated by the per neuron charge pumps are applied to two capacitive terminals. These capacitors allow parallel and local bidirectional weight updates. The drain and the source of Mr are pulled to vdd to minimize the voltage across internal parasitic capacitors (see Fig. 2.1 and 6.3). This four terminal FGT synapse exhibits memristor-like behavior in the learning mode in a sense that the conductance of both devices can be altered by applying an appropriate voltage (that exceeds the programming threshold) across the terminals. The memristor is addressed later in Chapter 7.

To increase the synaptic weight electrons must be added to the floating gate and subtracted from it to decrease the weight. This can be arranged by careful selection of the ratios of the capacitances $Cpre$, $Cpost$ and the gate capacitance of Mr so that tunneling occurs always across $Cpre$. The dominance of $Cpost$ over the total capacitance $Ctot$ helps Vfg to more closely follow the programming voltage applied to $Cpost$. In order to see how the gate current I_g through $Cpre$ affects the charge (or weight) at the floating node during tunneling, Eq. 2.1 is rewritten by noting that the charge Q is simply the time integral of I_g:

$$\Delta Vfg = \frac{I_g t}{C_{tot}}, \tag{6.2}$$

It is further assumed that the direct band-to-band tunneling is negligible, the barrier is triangular and the classical model for tunneling holds, so that I_g can be defined as:

$$I_g = I_0 e^{-\frac{V_f}{V_{ox}}}, \tag{6.3}$$

where I_0 is an empirically measured parameter, V_f is a constant that depends on the oxide thickness and V_{ox} is the voltage across the dielectric. This is a simplified version of Eq. 2.4. It can be observed that ΔVfg depends linearly on time but exponentially on the voltage across capacitor $Cpre$.

The shape of the biphasic tunneling pulse is the key for performing the weight updates according to the STDP rule. The idea is that an individual biphasic pulse delivered by a postsynaptic or a presynaptic neuron does not induce gate current (V_{ox} stays below the programming threshold), but when pulses occur simultaneously, their superimposed voltage magnitude exceeds the threshold. Fig. 6.9 illustrates the principle of weight updates based on biphasic programming pulses. Note that smaller time difference between the pulses leads to a larger peak magnitude of a $post - pre$ (and $Vfg - pre$). When the separation between pre and $post$ pulses starts to increase from zero, the amount of tunneling first increases and then starts to diminish.

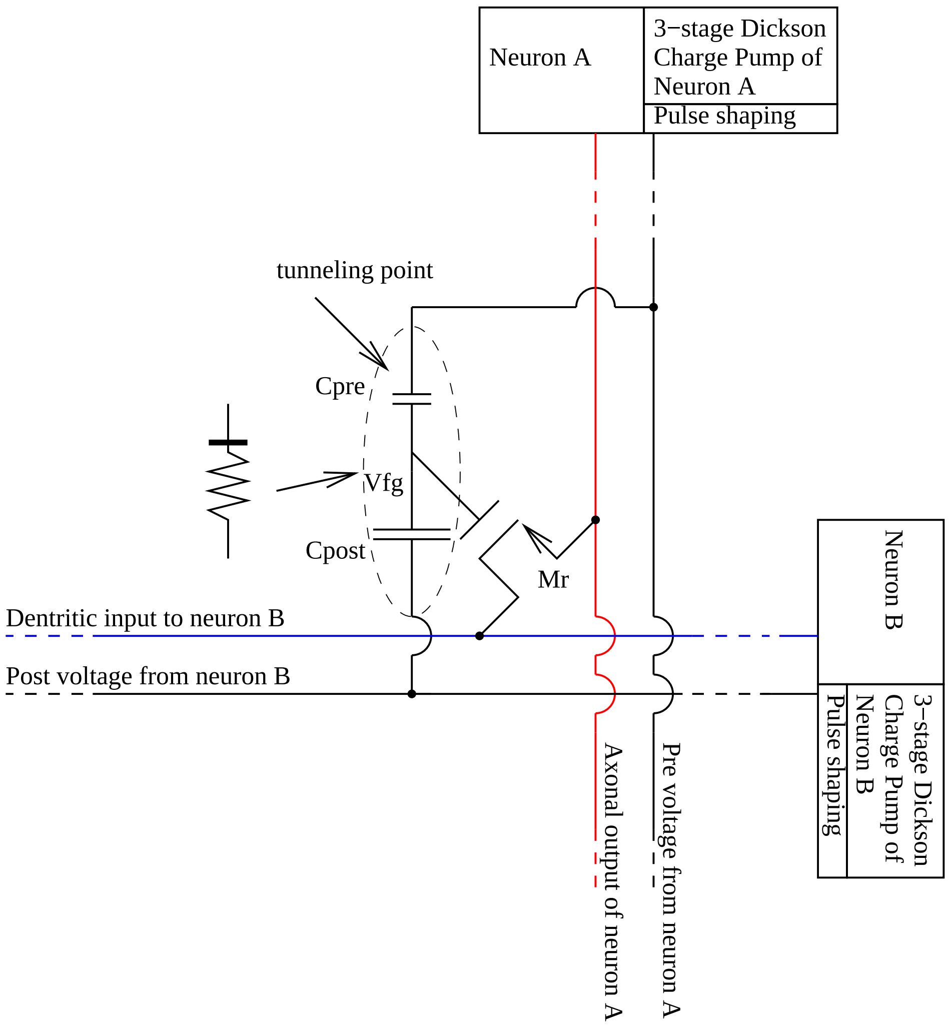

Neuron A
3−stage Dickson Charge Pump of Neuron A
Pulse shaping
tunneling point
Cpre
Vfg
Cpost
Mr
Dentritic input to neuron B
Post voltage from neuron B
Neuron B
3−stage Dickson Charge Pump of Neuron B
Pulse shaping
Axonal output of neuron A
Pre voltage from neuron A

Figure 6.7: The FGT synapse consists of one transistor with two capacitive connections. Capacitors are used to implement STDP weight updates in learning mode. The weight is stored as a charge on the floating gate and can be modified by applying simultaneous high voltage pulses on *Cpre* and *Cpost* during the learning mode. Capacitive ratios are selected so that tunneling occurs across *Cpre*. In the communication mode, the FGT synapse acts as current source that converts the voltage spike to a current pulse depending on the channel conductance defined by the stored charge.

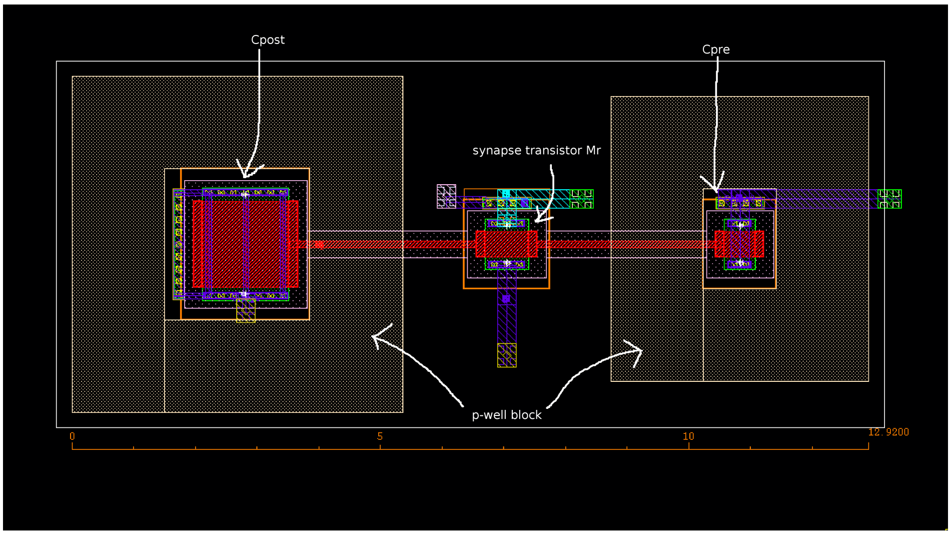


Figure 6.8: The layout of the FGT synapse with associated pmos well-capacitors. Each device is located in a separate n-well. N-wells are deposited sparsely and capacitors are surrounded by a p-well block for improved isolation. This is the reason why (especially single poly) floating-gate structures require a relatively large area although the devices themselves are quite small.

This can be observed from Fig. 6.10, where the difference in charge ΔQ is obtained by using relation

$$\Delta Q \propto \int e^{post(t)-pre(t)} dt \tag{6.4}$$

and by normalizing the result. The signal *post* refers to a biphasic high voltage pulse applied to $Cpost$, i.e. the pulse generated by the postsynaptic neuron. The signal *pre* is the pulse generated by the presynaptic neuron, respectively.

By examining equations 6.2 and 6.3, it can be seen that due to the exponential relationship, the magnitude of V_{ox} has a stronger effect on the accumulated charge than the tunneling time (linearly related). Consequently, a stronger weight modification is expected for the lower case in Fig. 6.9. Therefore, the method realizes a good approximation of STDP rule. In fact, the used approximation is even more robust (at least in hardware implementations) than the commonly used STDP rule with exponential decays. This is because, unlike original STDP, it provides a smooth transition as a function of the time difference between pulses and thus avoids the problem that small timing errors may cause large weight updates with wrong polarity.

In reality, due to the capacitive division, the magnitude of Vfg is smaller than *post* and consequently $post - pre$ is not directly the effective oxide

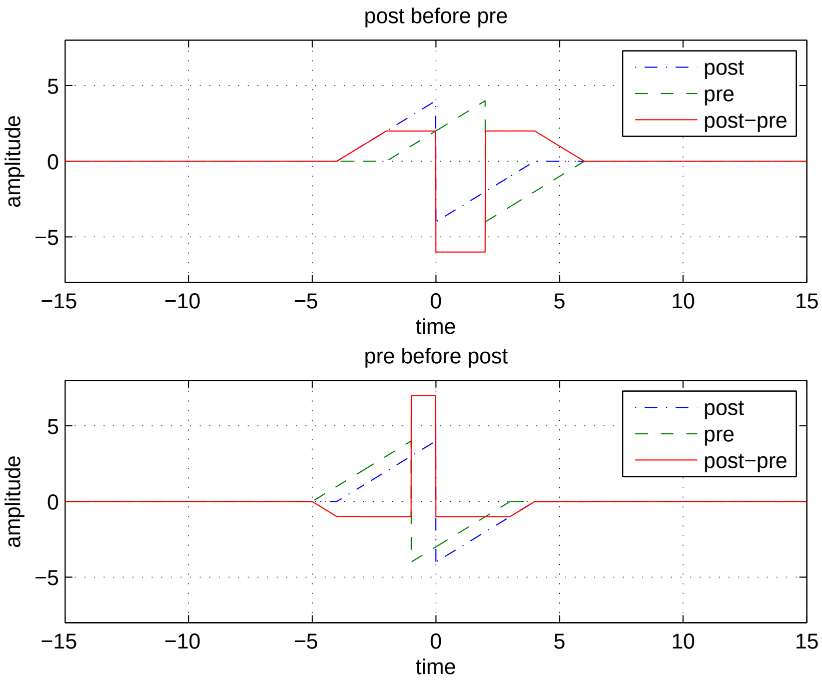


Figure 6.9: Biphasic high voltage pulses *pre* and *post* applied to respective capacitors $Cpre$ and $Cpost$ with two different delays shows how the order of *pre* and *post* determines the polarity and the time difference the amplitude of resultant voltage $(post - pre)$ across $Cpre$.

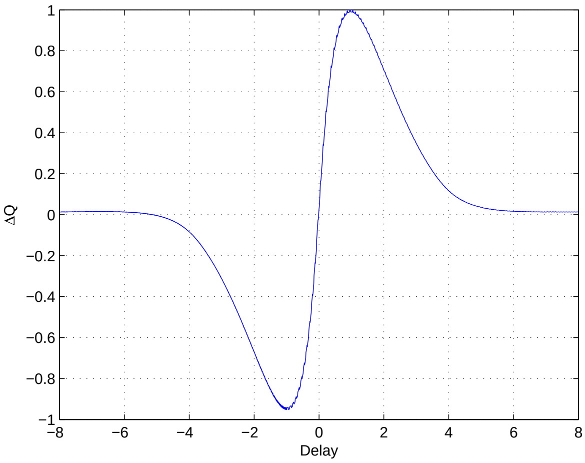


Figure 6.10: Normalized change of charge as a function of delay between *post* and *pre* pulses.

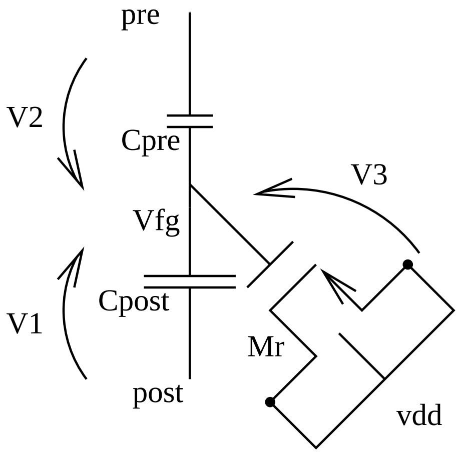


Figure 6.11: The oxide voltages $V1$, $V2$ and $V3$ of FGT synapse in learning mode.

voltage (that is $V_{ox} = |Vfg - pre|$). To avoid unintentional weight updates, the potential difference across all capacitors connected to the floating gate should be kept minimal during the communication mode and in the case where *pre* and *post* pulses do not overlap. On the other hand, overlapping pulses should evoke intentional weight modifications.

Potential differences across gate oxides are defined as: $V1 = post - Vfg$, $V2 = pre - Vfg$ and $V3 = Mr_{d,s,b} - Vfg$ (see Fig. 6.11). $Mr_{d,s,b}$ denotes the drain/source/bulk voltages of synapse transistor Mr which are all pulled to *vdd* during programming. Figure 6.12 shows a simulation of the synapse of Fig. 6.7 that aims to assess unintentional tunneling through transistor Mr. Distinct biphasic high voltage pulses are applied to $Cpre$ and $Cpost$. All devices are pmos transistors (including capacitors) with the nominal values of $Cpre$=1.42 fF, $Cpost$=12.25 fF and a total gate capacitance for Mr of Cgg=1.97 fF. These values correspond to W/L ratios 0.5/0.4, 1.4/1.4 and 0.7/0.4, respectively (all dimensions in μm). In this particular example, the minimum effective voltage across the gate oxide that triggers F-N tunneling is estimated to be 3.7 V based on information given in [29] (t_{ox} x 6.4 x 10^8 V/m). A tunneling voltage this low requires very long tunneling times, but represents a rough artificial threshold voltage which should not be crossed unintentionally.

From Fig. 6.12 it can be observed that a distinct biphasic high voltage pulse applied to $Cpre$ or $Cpost$ can not alone induce tunneling (absolute values <3.7 V). As $V1$, $V2$ and $V3$ are referred to Vfg, the initial charge at the floating node affects their magnitude directly. In the simulation plotted in Fig 6.12, the initial floating-gate voltage was set to 2.5 V (Vdd). A lower initial voltage shifts the curves upward so that $V2$ meets the upper bound

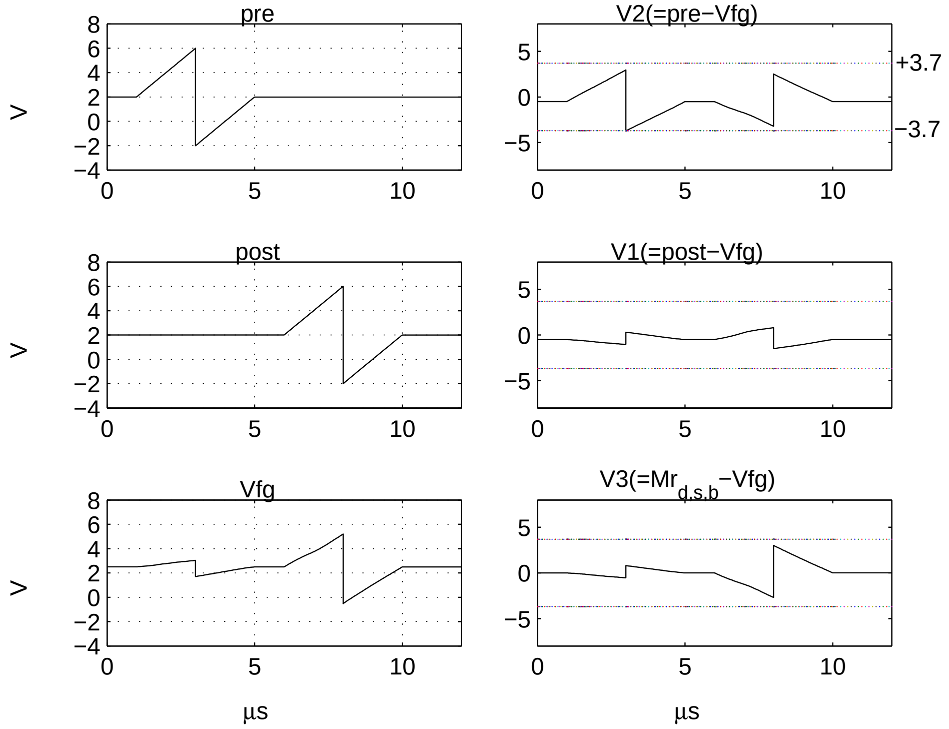


Figure 6.12: A biphasic high voltage pulse applied to either *Cpre* or *Cpost* can not alone induce tunneling. Potential difference across all capacitors stays below 3.7 V, which is estimated to be the minimum voltage for F-N tunneling in this process.

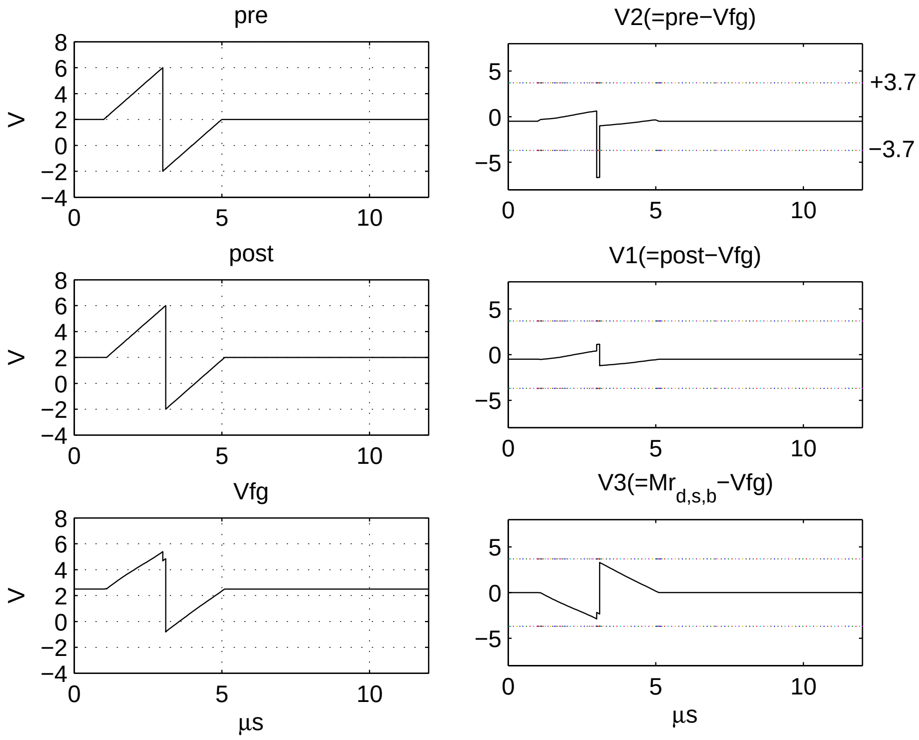


Figure 6.13: A transient simulation where *pre* pulse comes 100 ns before *post*. A large negative voltage is formed across $Cpre$ and electrons are added to the floating-gate. Consequently, the weight of the synapse is increased.

with the initial voltage of 1.5 V. This is not a problem since the synapses are intended to work in subthreshold region which means that Vfg is between 2 V and 2.5 V.

Figure 6.13 shows a simulation of overlapping high voltage pulses with a small delay between *pre* and *post*. Because of this, tunneling occurs ($| V2 |> 3.7$ V) and the weight of the synapse is increased. Similarly, if the *post* pulse comes before *pre*, tunneling takes place in a different direction. Simultaneous pulses do not induce tunneling.

In practical circuits, there is no need to use biphasic triangular pulses used in previous simulations, since the shape of the pulse is not very important. Thus, pulse shapes with e-fold increase and decrease characteristic of a step response of an RC circuit can be used without essential changes in the functionality of the method.

Design Choices for Generating the High Voltage Biphasic Pulses

From Fig. 6.12 and 6.13 it can be seen that *pre* and *post* range from -2 V to 6 V resulting a peak-to-peak value of 8 V. However, during the design phase, it turned out that due to technological limitations discussed in Chapter 3, the application of negative tunneling voltages would be difficult to realize in the 65 nm general purpose process. Consequently, a unipolar supply was chosen.

The second challenge was to properly set the DC-levels of the tunneling voltages so that intentional programming was possible yet the probability of unintentional tunneling could be minimized at the same time. The fact that the DC-level of a high voltage pulse lies between the supply rails poses an additional design challenge. Setting the supply voltages of the neuron and synapse to 2 V (*vss*) and 4.5 V (*vdd*), and the DC-level of tunneling voltages (*pre* and *post*) to 4 V were found to be a satisfactory compromise that allowed swings of ± 4 V around the bias point.

In the beginning, the aim was to have one common high voltage source, either on-chip or off-chip, that delivers high voltages for synapses through switches on demand. However, the implementation of high voltage floating switches that can tolerate voltages up to eight volts would be expensive in terms of area with the 65 nm process. This is because sufficient devices (HV transistors) are not provided by the general purpose design kit. Indeed, it is possible to construct a high voltage tolerant switch using only standard transistors for example by following the guidelines presented in [11]. The principle is shown in Fig. 6.14. The idea is to keep the drain-source voltages low by stacking multiple devices. However, this approach was quickly abandoned because it resulted in very complex circuits. 8 V tolerance requires at least quadruple-cascodes assuming 2.5 V nominal Vdd tolerance for thick oxide devices. Furthermore, setting the DC-level properly was particularly challenging by using this approach.

The next alternative was to consider including a charge pump in each neuron in order to avoid high voltage tolerant switches. It was realized that the ramp-up time of the charge pump could be used to generate the desired shape for the high voltage pulses required for programming. Furthermore, the desired DC-level needed in the communication mode could be achieved simply by closing part of the pumping stages. Putting these principles into practice resulted in implementation presented in Fig. 6.15. That is, each neuron is equipped with a three-stage Dickson type charge pump to provide biphasic high voltage pulses during the learning mode.

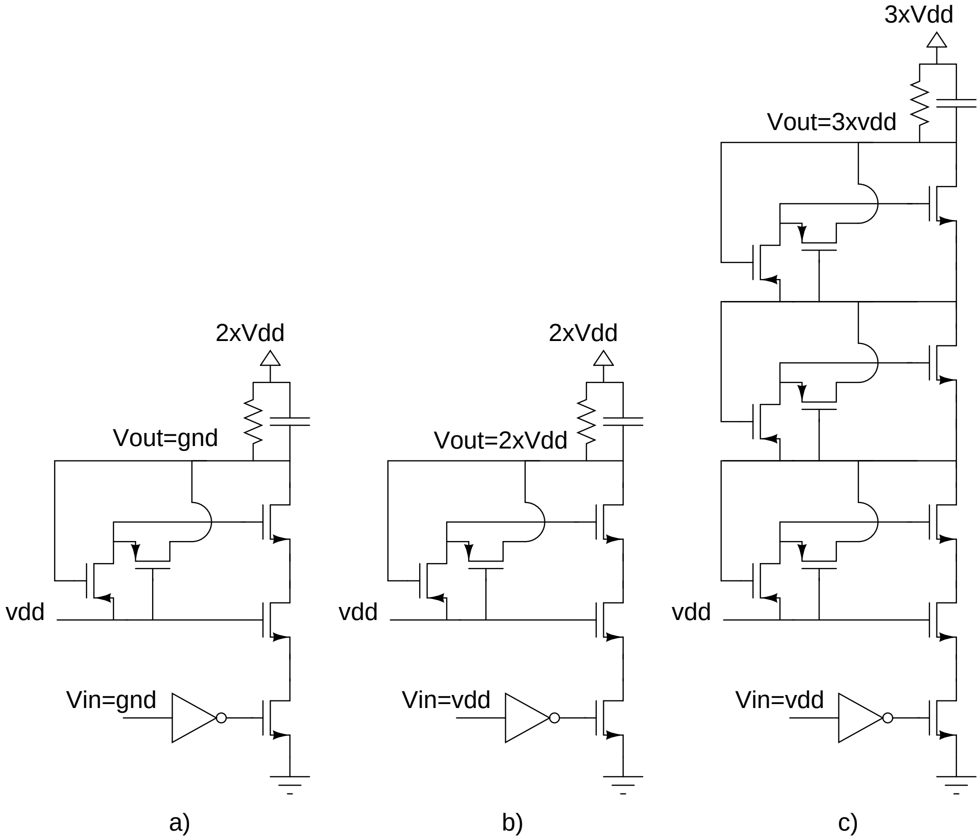


Figure 6.14: The principle of increasing the voltage tolerance of circuits using cascode transistors. Two pmos transistors comprise a "soft" switch that drives the nmos cascode transistor. A dual-cascode doubles the tolerated voltage as shown in a) steady state low and b) steady state high. A quadruple-cascode triples the tolerated voltage.

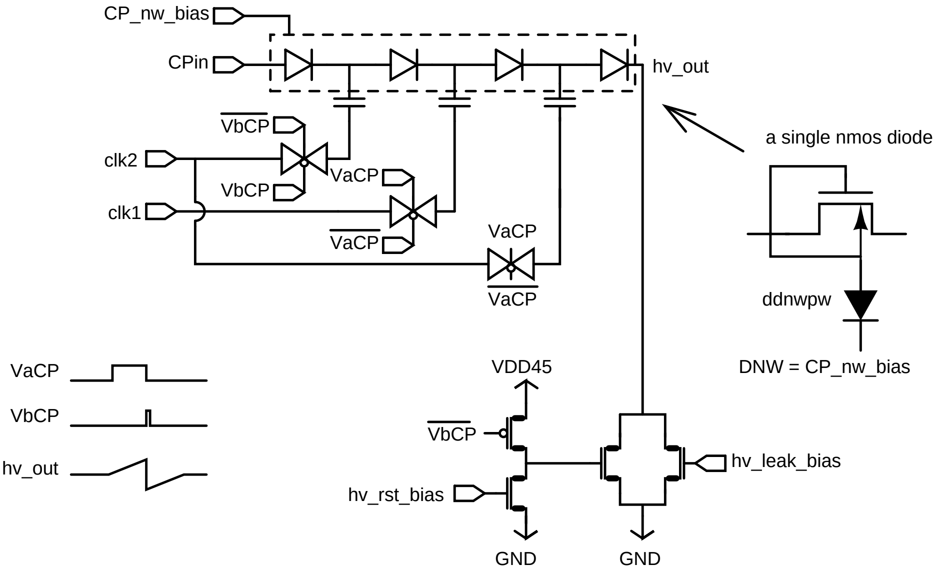


Figure 6.15: Each neuron is equipped with three stage Dickson type charge pump for generating biphasic high voltage pulses that are used for weight modifications during the learning mode. Two non-overlapping clock signals driving the charge pump are gated so that only the first stage is active in the communication mode. In the learning mode, also the later stages are active when Va is high. The circuitry for generating the control signals $VaCP$ and $VbCP$ is shown in Fig. 6.3. Transistors with bold horizontal lines are drift transistors that can tolerate higher drain-source voltages than standard transistors. Nmos diodes are in separate p-wells inside a common deep n-well (DNW). *ddnwpw* is a diode that is formed between each p-well and DNW.

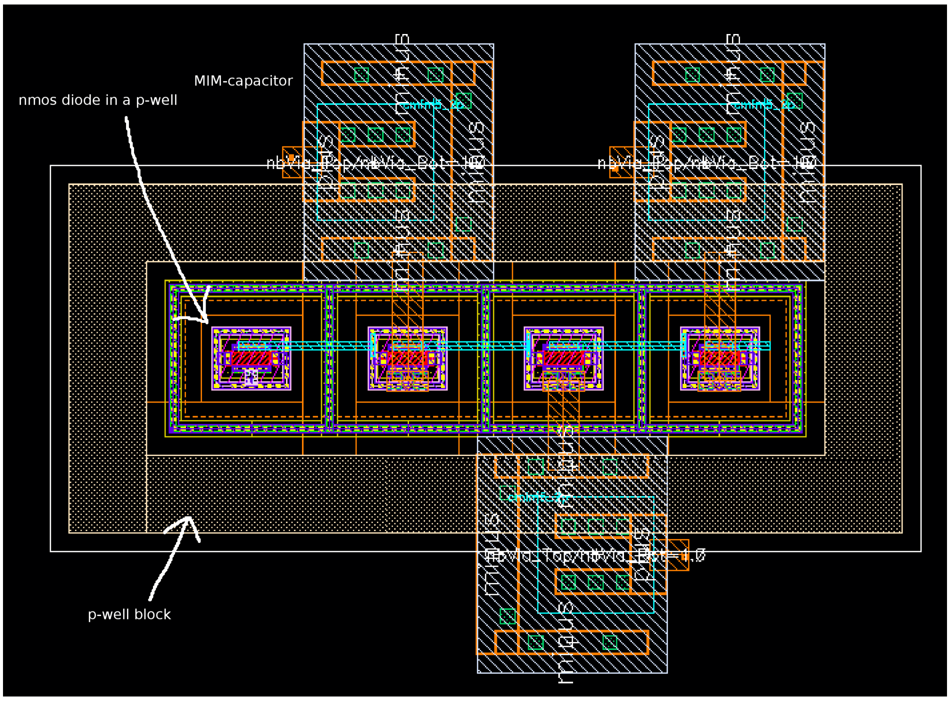


Figure 6.16: The layout of the three-stage Dickson type charge pump. Each diode connected nmos transistor is located in a separate p-well inside a common n-well (DNW). Each p-well is surrounded by a guard-ring and the common n-well is surrounded by a p-well block for improved lateral isolation.

Charge Pump

A Dickson-type charge pump requires one diode per pumping stage with one additional diode for input. The pumping capacitors raise the output voltage gradually when clocked with an appropriate frequency. The corresponding layout of the diodes and Metal-Insulator-Metal-capacitor (MIM-capacitors) is shown in Fig. 6.16. The function of the charge pump in both communication and learning modes is explained next.

In the communication mode, only the first stage of the Dickson charge pump is active ($VaCP$ and $VbCP$ are low) and the output voltage $Vout$ resides in the middle of the voltage range. The size of the pumping capacitor, the clock frequency and the input voltage of the charge pump influence the efficiency of the charge pump. Furthermore, the maximum output voltage and the rise time depend on the number of pumping stages and the load capacitance. Because the load is (almost) purely capacitive, the DC-level must be set by allowing a small leakage current with bias hv_leak_bias, otherwise the charge pump will drive $Vout$ slowly higher and higher. The inactive pumping stages introduce some voltage drop because of the forward biased diodes. This is compensated by sizing the charge pump so that the first stage can drive the load to a desired voltage despite the inactive stages that lower the output voltage. The size of the pumping capacitors, input voltage $CPin$, and load capacitance are fixed in the design phase. Only the clock frequency, hv_leak_bias and CP_nw_bias are accessible parameters after fabrication and thus are used to set the DC-level and the deep n-well bias of the charge pump.

In the learning mode, also the later stages of the charge pump are clocked and the output voltage is gradually driven to peak value. The active period of the charge pump lasts as long as Va stays high. Next, when Vb goes active, $Vout$ is pulled down to ground by a drift transistor that tolerates high drain-source voltages. More specifically, a drift transistor allows the electrical field to be spread into the n-well and p-well so that the effective voltage applied to the channel and the gate oxide is lowered[1]. The strength of the pull down transistor is controlled by biasing the overdrive voltage with hv_rst_bias.

All pumping stages are inactive when Vb is high to assist pull down so that the drift transistor does not have to be a much stronger driver than the charge pump. After Vb goes inactive, the first stage of the charge pump lifts $Vout$ to the mid voltage. The input voltage for charge pump $CPin$ is also increased when Va is high (the uppermost diode connected transistor is bypassed) and respectively reset by $VbCP$, see Figs. 6.3 and 6.5. Increased $CPin$ allows the charge pump to achieve higher maximum voltage without increasing the number of pumping stages whereas gradually increasing $CPin$ after reset slows down the rise time to mid voltage balancing the rise times from different initial values. A transient simulation of the charge pump is shown in Fig 6.17.

Since the learning is local, each neuron needs to be accompanied by a charge pump. The charge pump uses MIM capacitors that are located above the wiring layers allocated for intraneuronal communication. Because the rise time of $Vout$ is heavily influenced by the capacitive load at the output, dummy synapses were added on both input and output layers so that the capacitive load of charge pumps is equalized (see Fig. 6.18). Otherwise, neurons in the output layer would have to drive a load capacitance of N x $Cpost$, which is much larger than the load capacitance of neurons in the input layer N x $Cpre$, where N is the number of neurons in a single layer. The sensitivity to the load capacitance is one issue that needs to be addressed if the same approach is extended to larger arrays.

[1] Device does not require any additional processing or masks.

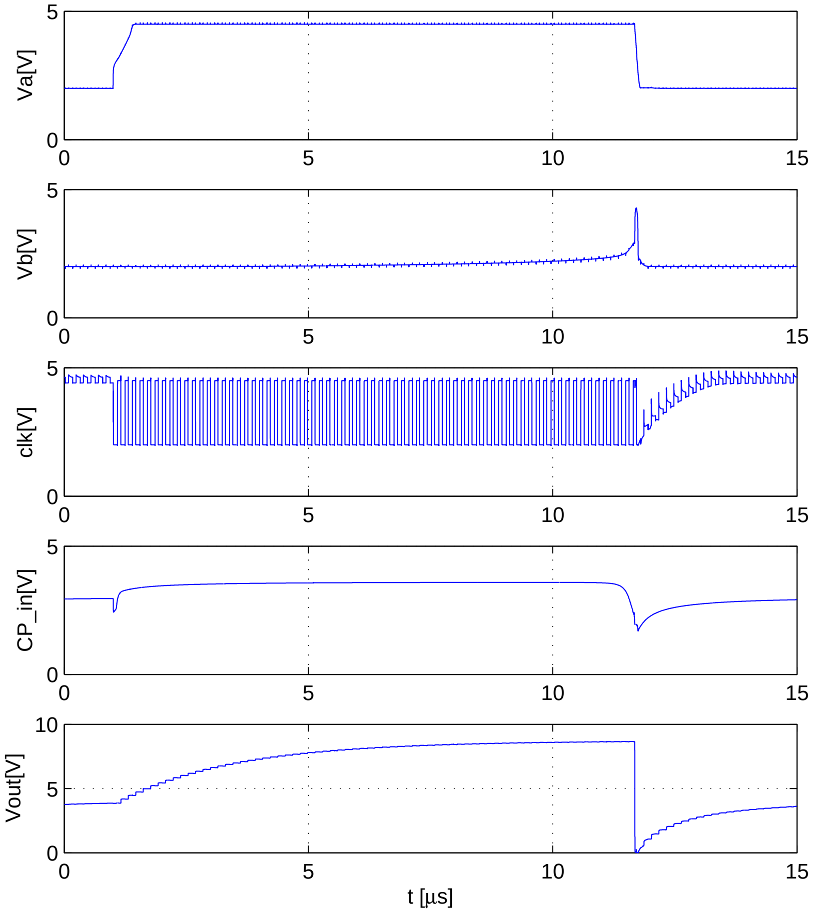


Figure 6.17: A transient simulation of the charge pump showing how the on-chip biphasic pulses are generated in the learning mode. Stage one keeps the output $Vout$ at approximately 4 V (DC-level) and thus is active in both the communication and the learning modes. Stages two and three are clocked only when Va is high in the communication mode. Stage one restores the DC-level after Vb has reset the output. The input of the charge pump $CPin$ is not constant but depends on the control signals Va and Vb so that the pump achieves higher peak voltage and the rise times (from DC to max and from ground to DC) are balanced. In this example, a clock frequency of 10 MHz (the clk signal is plotted after the t-gate) and load capacitance of 200 fF were used.

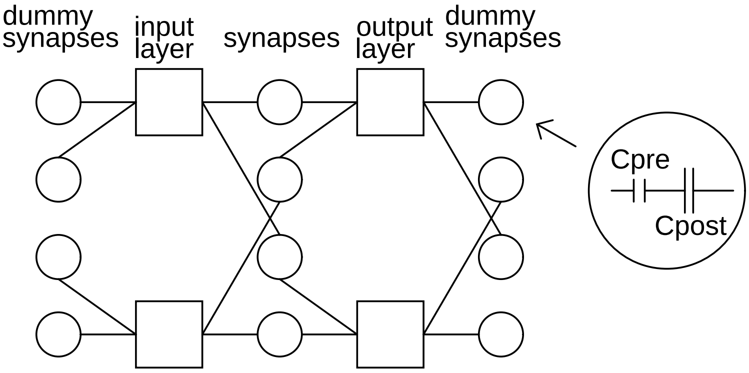


Figure 6.18: An example of a fully connected 2x2 SNN. Dummy synapses are added to both the input and the output layer so that each charge pump drives a capacitive load of equal size. Otherwise the capacitive load for charge pumps in the output layer would be $\frac{Cpost}{Cpre}$ higher than for charge pumps in the input layer.

6.1.3 Experimental Results of the Synapse

The concept of weight modification based on the time difference of high voltage biphasic pulses is demonstrated with the current reference circuit presented earlier in Section 4.2, because the SNN chip does not contain separate test synapses. The programmable current reference is suitable for this purpose, although it was not originally designed with this functionality in mind. The design is redrawn in Fig. 6.19 to clarify the test setup.

Two instances of off-chip high voltage biphasic pulse generator shown in Fig. 6.20 were used to mimic the charge pump based pulse generators in the SNN chip. The time difference of pulses is controlled with a pattern generator hosted by a pc. The maximum output voltage of the pattern generator is 5 V, thus inverters (4049) are used as level shifters to generate appropriate control voltages for the actual pulse generator circuit. Image captures from an oscilloscope shown in Figs. 6.21 and 6.22 show the generated pulse pairs applied to capacitors *Cpre* and *Cpost* of the on-chip current reference circuit. There is some inter-pulse variation in the generated maximum voltage (biphase_max) due to nonidealities, such as noise and jitter. This directly affects the effective tunneling voltage across the tunneling junction. To improve the reliability of measurements at least 10 pulses were applied per tunneling cycle to average out the effect of the variance between consecutive pulses.

The source voltage of $M1$ (vdd_fg) was raised to 4.5 V and the drain current ($Iout$) was measured by connecting a picoammeter between to_pad and ground. The obtained measurement results presented in Figs. 6.23 and

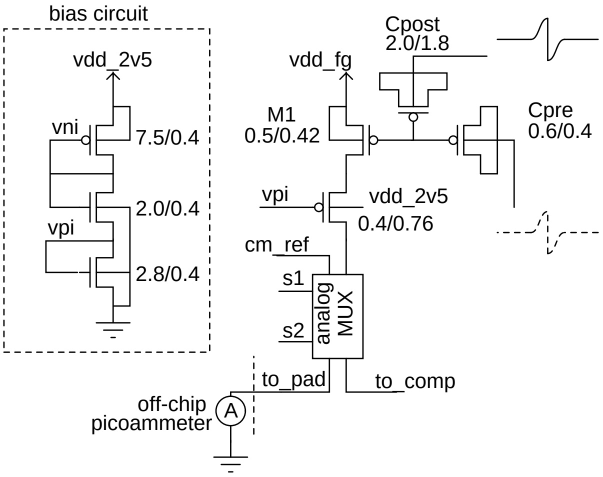


Figure 6.19: A current reference circuit designed in 90 nm standard CMOS was used to demonstrate the weight modification concept based on biphasic pulses. An analog MUX is configured so that the drain current of $M1$ (Iout) flows to the ground through the picoammeter. The direction of weight update is determined by the order of biphasic pulses, whereas the strength is determined by the time delay between the pulses applied to $Cpre$ and $Cpost$ respectively.

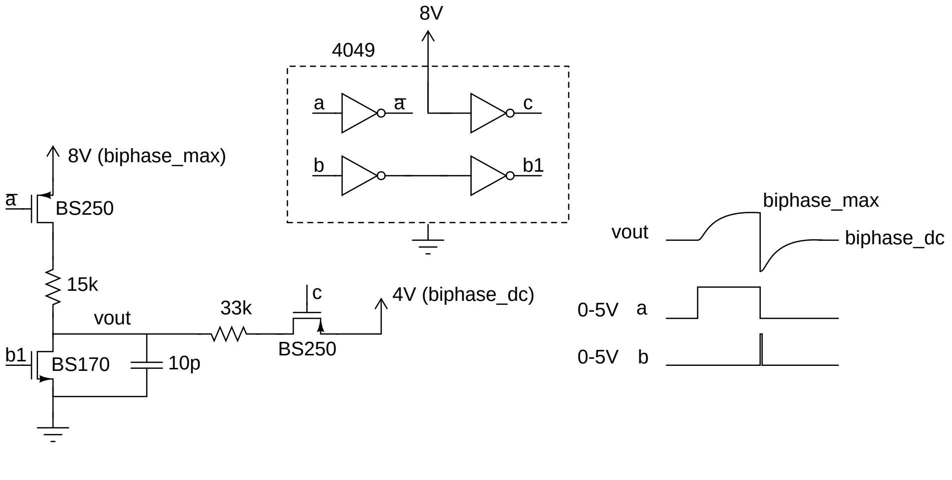
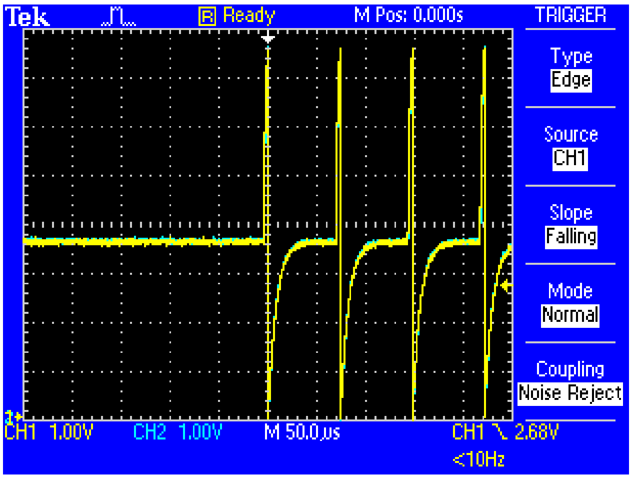


Figure 6.20: An off-chip circuit that generates high voltage biphasic pulses.



M Pos: 0.000s

Type: Edge

Source: CH1

Slope: Falling

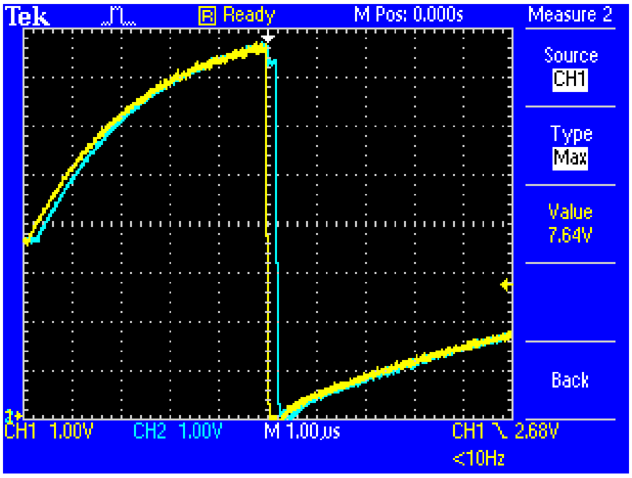
Mode: Normal

Coupling: Noise Reject

CH1 1.00V CH2 1.00V M 50.0µs CH1 ↘ 2.68V

<10Hz

Figure 6.21: The oscilloscope image shows four pairs of consecutive biphasic pulses generated by an off-chip pulse generator shown in Fig. 6.20. In the measurements at least 10 pulses are applied in each tunneling cycle to average out the small variations between consecutive pulses.



M Pos: 0.000s

Source: CH1

Type: Max

Value: 7.64V

CH1 1.00V CH2 1.00V M 1.00µs CH1 ↘ 2.68V

<10Hz

Figure 6.22: The oscilloscope image shows one pair of biphasic pulses (*pre* in CH1 and *post* in CH2) generated by an off-chip pulse generator shown in Fig. 6.20. The time delay between the pulses is 200 ns and the peak-to-peak voltage is approximately 7.8 V.

6.24 verify that the biphasic pulses can be used to realize the STDP learning rule as hypothesized. The change in $Iout$ after eight tunneling cycles reaches a maximum with a 300 ns delay in Fig. 6.23 and a minimum with a 200 ns delay in Fig. 6.24, respectively. This consistent with the theory presented in Fig. 6.10.

Simultaneous pulses did not cause measurable changes in $Iout$ (1500 pulse pairs applied) and increased time difference between biphasic pulses decreases the strength of weight modification as predicted. From Eq. 2.3 it follows that the amount of required pulse pairs to achieve a desired change in $Iout$ depends on the initial charge stored on the floating gate. In these examples, the initial current is approximately 1 μA, which means that floating-gate voltage V_fg is approximately 4 V. That is why more pulse pairs are required to further decrease the current than are needed for increasing the current. This imbalance can be compensated to some extent by modifying the pulse parameters *biphase_DC* and *biphase_max* shown in Fig. 6.20. Fig. 6.25 demonstrates the effect of compensation with two parameter sets (par1 and par2). With parameter set 1 (par1) the positive slope is steeper than the negative slope despite the ratio of applied pulse pairs is 10 to 200 per tunneling cycle. The parameter set 2 (par2) results in equal slopes although the ratio of applied pulse pairs is significantly smaller (20 to 50 per tunneling cycle).

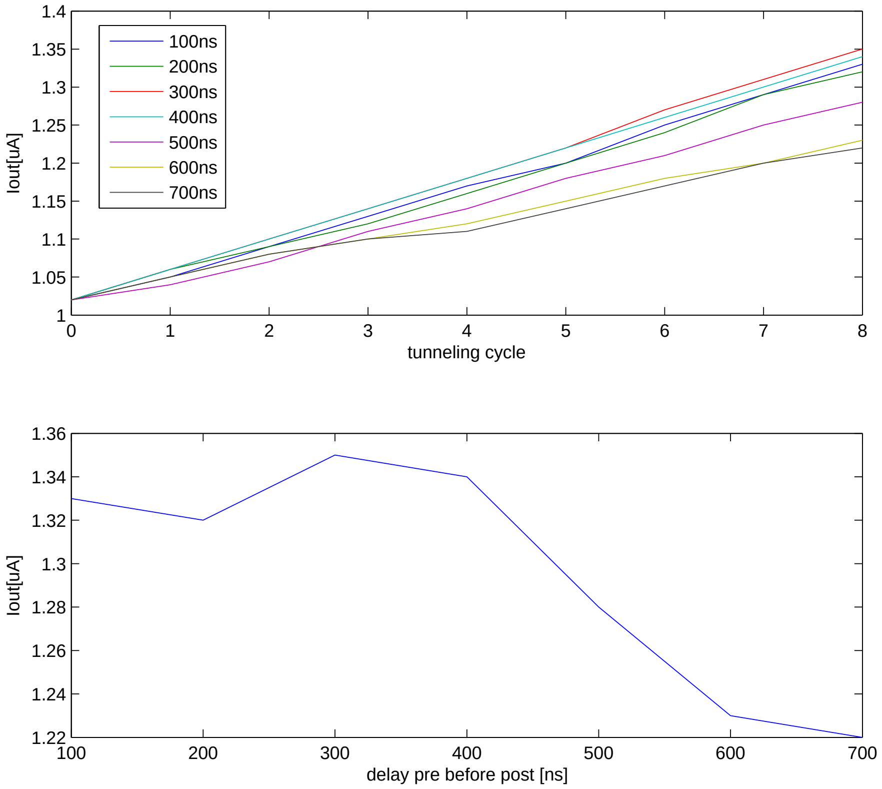


Figure 6.23: The drain current (Iout) of the p-type floating gate transistor increases when the biphasic high voltage pulse is applied to $Cpre$ before $Cpost$. In this experiment, ten biphasic pulses per one tunneling cycle were applied with varying time delay ($Iout$ initialized to 1 μA between each run with different delay). The time delay between the applied pulses (shown in the legend) influences on the change in $Iout$ per one tunneling cycle according to the STDP rule (upper figure). The resulting $Iout$ after applying eight consecutive tunneling cycles as a function of time delay is shown in the lower figure. As $Iout$ peaks with 300 ns delay, the measured data coincides well with the data predicted in Fig. 6.10. The key parameters of the biphasic pulses used in this experiment were: vdd_fg=4.5 V, $biphase\_max$=7.20 V, $biphase\_dc$=3.98 V.

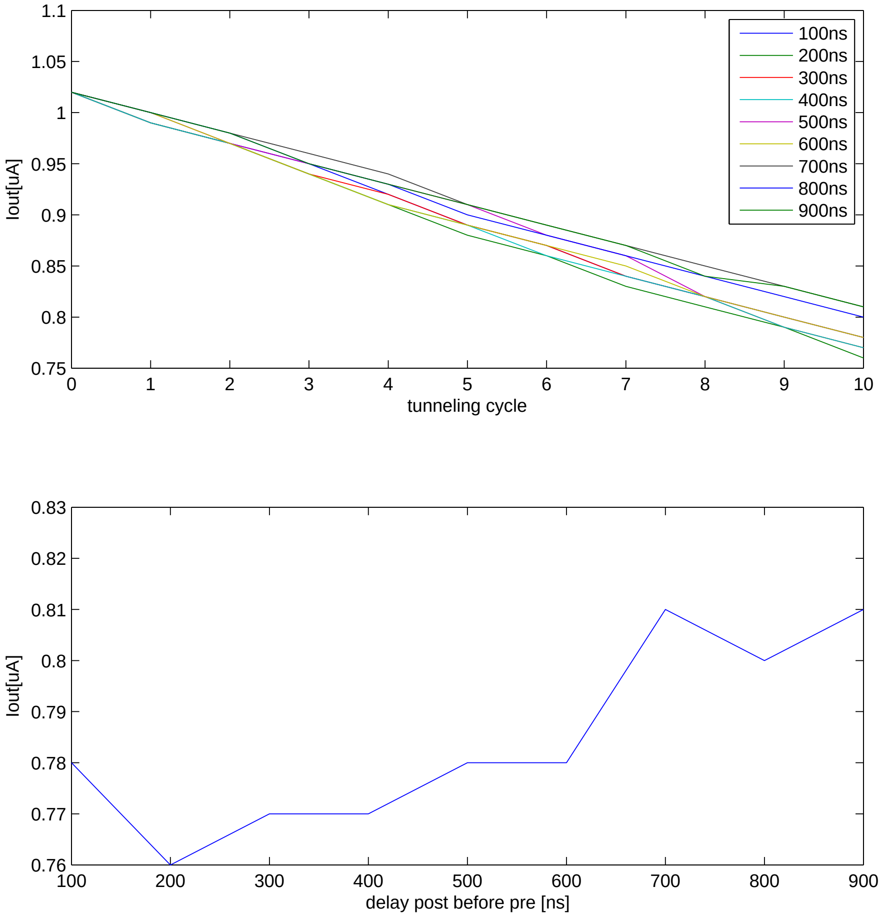


Figure 6.24: The drain current (Iout) of the p-type floating gate transistor decreases when the biphasic high voltage pulse is applied to $Cpost$ before $Cpre$. In this experiment, 200 biphasic pulses per one tunneling cycle were applied with varying time delay ($Iout$ initialized to 1 μA between each run with different delay). The time delay between the applied pulses (shown in the legend) influences on the change in $Iout$ per one tunneling cycle according to the STDP rule (upper figure). The resulting $Iout$ after applying ten consecutive tunneling cycles as a function of time delay is shown in the lower figure. As $Iout$ has a minimum at 200 ns delay, the measured data coincides well with the data predicted in Fig. 6.10. The key parameters of the biphasic pulses used in this experiment were: vdd_fg=4.5 V, $biphase\_max$=7.20 V, $biphase\_dc$=3.98 V.

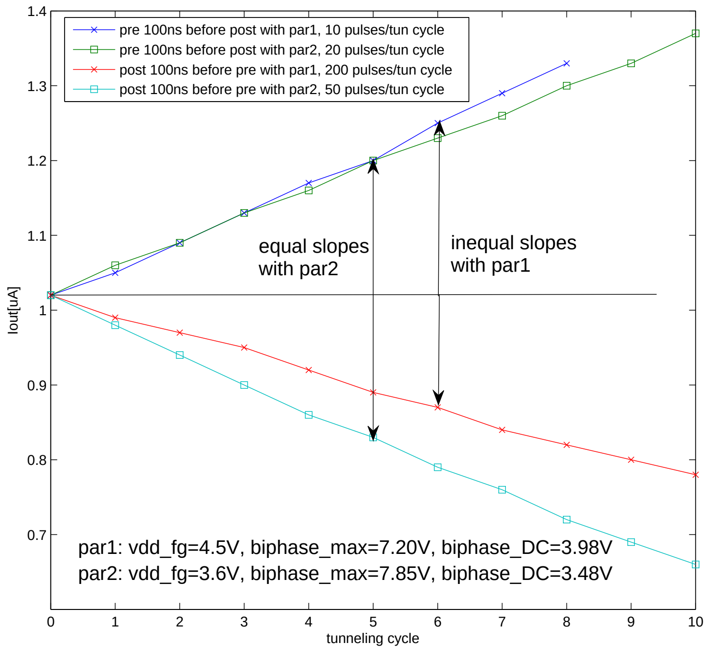


Figure 6.25: The accumulated charge per one tunneling cycle can be tuned by changing the parameters of the biphasic pulse and the terminal voltages of the floating gate transistor. Here, the supply voltage (vdd_fg), $biphase\_max$ and $biphase\_DC$ are modified so that the positive slope remains nearly unchanged although the pulse count per tunneling cycle is doubled, whereas the negative slope increase although the pulse count per tunneling cycle is only one fourth of the initial. That is, the changes in Iout in both directions are more balanced with parameter set 2.

6.1.4 Experimental Results of the Charge Pump

The SNN array prototype chip includes an IO-pin to monitor the high voltage output of the separate test neuron. It allows verifying if the charge pump is able to generate a correct biphasic waveform in learning mode. Observations can also be made if the response to changes in different on-chip bias voltages meets expectations. Fig. 6.26 shows the output of the charge pump (channel 2) when the chip is configured to the learning mode and the test neuron is addressed (channel 1). Control signal $vfall$ is set so that the charge pump has plenty of time to run with all pumping stages active (output starts to saturate). This plot revealed a problem with the charge pump: With nominal bias and supply voltages the dc-level is about 2.2 V instead of expected 4 V and the maximum voltage is about 5.8 V instead of expected 8 V.

Since the designed charge pump is intended to drive only capacitive loads, it can not generate targeted voltage levels to resistive loads that sink large currents. Therefore, the analysis of this problem was started by minimizing the leak current through the drift transistors by biasing the hv_leak_bias to 0 V, so that both of the drift transistors were cut-off. By doing so, the resistive load should be in the order of few $M\Omega$s assuming that the resistances in parallel with the probe (10 $M\Omega$ / 16 pF @ 10 x gain) are larger than that of the probe. However, the DC-level did not increase higher than 2.2 V even if the frequency of $clk1$ and $clk2$ was increased. On the other hand, the output fell to 0 V as it should do, when the clock circuit was biased into off-mode. The validity of the assumption concerning internal resistances was confirmed by changing the gain setting of the probe. Fig. 6.27 demonstrates the loss of performance when the gain setting of the probe is changed (1 $M\Omega$ / 95 pF @ 1 x gain). There is a dramatic drop in voltage levels, because of the limited drive capability. However, if the load resistance in the simulator is adjusted to match the measured DC-level of Fig. 6.26, the corresponding peak voltage is only 3.2 V which is substantially lower than the observed 5.8 V. Therefore, it is unlikely that the increased load resistance alone explains the observed problem. It should be noted that the probe is not loading the internally connected neurons, so that it can be assumed that the programming voltages are actually higher than those measured from the test neuron.

The next step was to try carefully change the supply voltages $vdd4v5$, vss, $vdd2v5$ and the on-chip biases generated by the on-chip DACs. The response to each of the adjustable parameters was as expected, but did not solve the problem. That is, none of the tested chips was able to achieve the targeted voltage range. Fig. 6.28 shows a typical hv pulse when the chip was configured for maximal output voltage. However, the achieved voltage levels 2.9/6.9 V are only about 1 V below the targeted values 4/8 V.

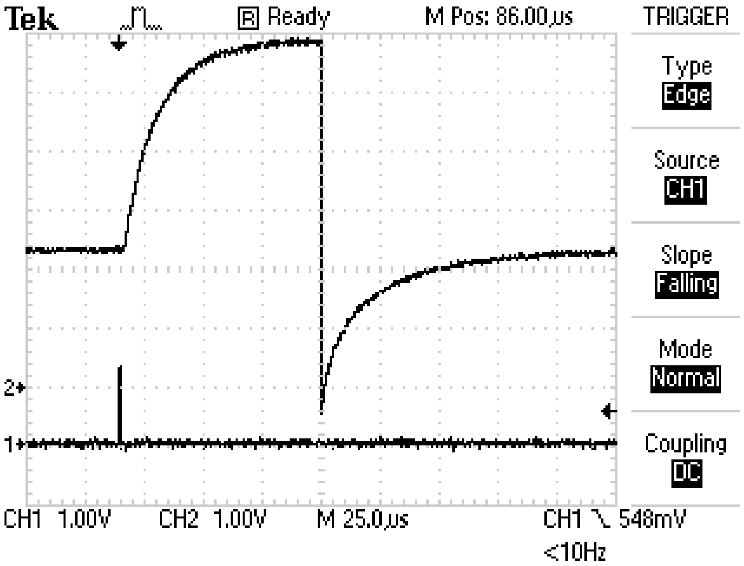


Figure 6.26: The oscilloscope image shows the biphasic pulse generated by the on-chip charge pump shown in Fig. 6.15. The shape of the pulse is correct, but the dc-level is about 2.2 V instead of expected 4 V. Consequently, the weight adaptation in the learning mode does not work as designed.

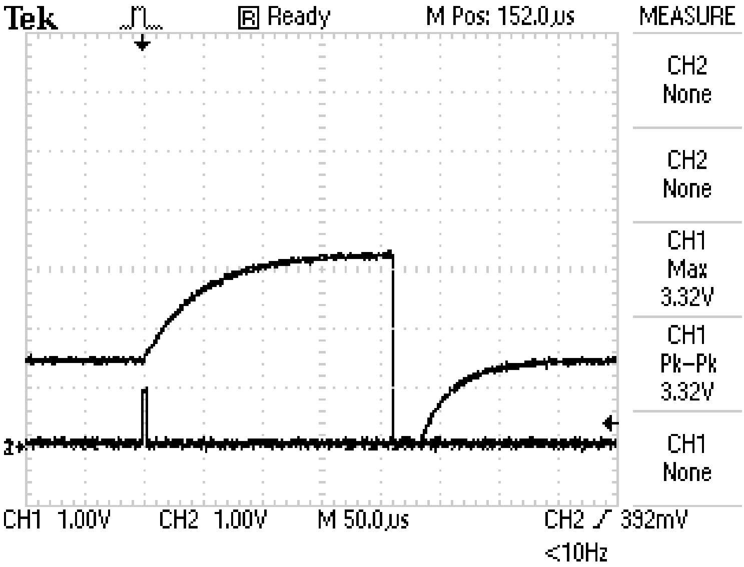


Figure 6.27: If the resistance of the load is too small the performance of the charge pump collapses, because it is not designed to drive difficult resistive loads.

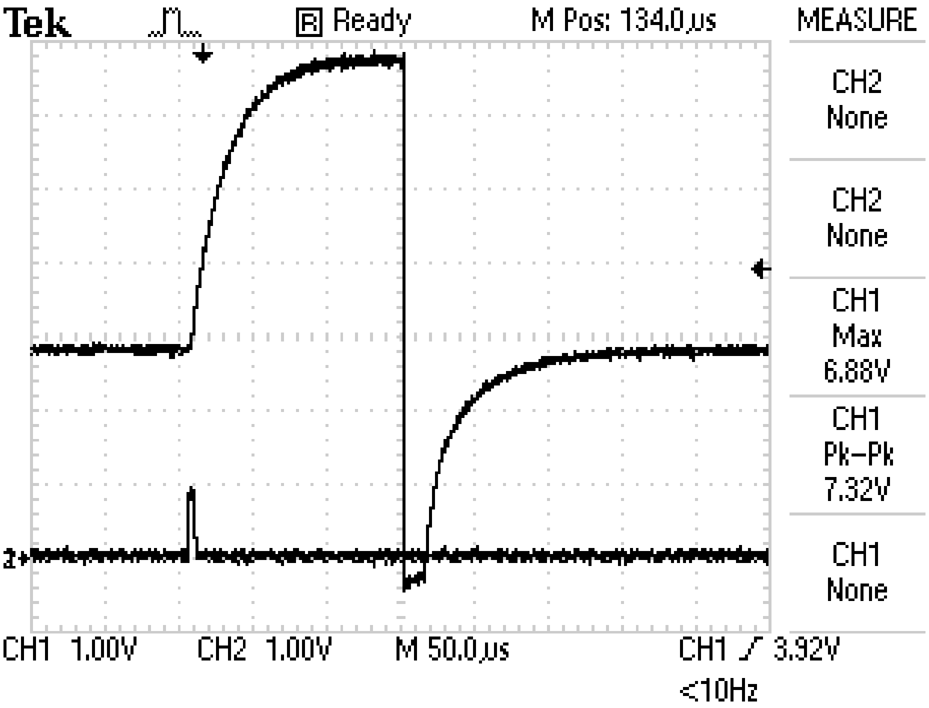


Figure 6.28: The oscilloscope image shows the typical biphasic pulse generated by the on-chip charge pump when configured for maximal output voltage. The dc-level is about 2.9 V and the maximum voltage about 6.9 V respectively.

Finally, the layout of the charge pump was checked for concealed defects. To help the analysis of the physical structure, a detailed cross-section of the implemented topology was drawn, with parasitic resistors and BJTs as shown in Fig. 6.29. The Fig. 6.29 helps to verify, that the latch-up is effectively prevented, since the deep n-well (DNW) is in the highest potential (8 V) and $nwgr > x > in$, so that the $Qp2$ is cut-off. However, there is a resistive path between consecutive stages through the p-well. The sheet resistance of Rpw is approximately 1.5 k$\Omega/\square$ and the distance is less than 1 μm, suggesting that this parasitic resistance could at least partly explain why the performance of the charge pump is lower than expected. The conclusion is that there is no single reason for the lowered DC-level. Instead, two or more, possibly very complex linear or non-linear, phenomena are responsible for the observed behavior, and unfortunately the identification and further analysis of these is nearly impossible with the current measurement setup.

6.1.5 Experiments with the Array

In the communication mode, the neurons in the input layer can be stimulated externally and the resulting spike counts monitored in the output layer. Because the measured dc-level and maximum voltage of the programming pulse were a little bit below the targeted values, the overlap of the programming pulses was increased respectively, so that the extended tunneling time could compensate for the decreased voltage across the tunneling junction. In this manner, the strength of the effective weight adaptation mechanism should remain nearly constant. With carefully adjusted bias voltages, supply voltages and AER parameters (e.g. delays in handshaking), it was possible to change the strength of synaptic weights on and off in both directions several times in a row. However, the setup was very sensitive and the amount of registered events varied a lot with the same input stimulus. An example of observed changes as a result of binary programming is given in table 6.1.

The table 6.1 is interpreted as follows. In the initial state no events are registered for any neuron in the output layer (column init.). Next, all synaptic connections are strengthened (excluding inhibitive connections) and the spike counts are read twice (R1 and R2). There is no programming in between the read cycles and the input stimulus is always the same. The first row shows the direction of programming, where "on" means that connections are strengthened and "off" that they are weakened respectively. The number of applied programming pulses was nine (the tunneling time was approximately 700 μs per pulse) and also the number of read pulses was nine. A key observation is that nearly all neurons fire spikes after the first programming cycle, stop firing after decreasing the synaptic weights and return into active state after reversing the programming direction again.

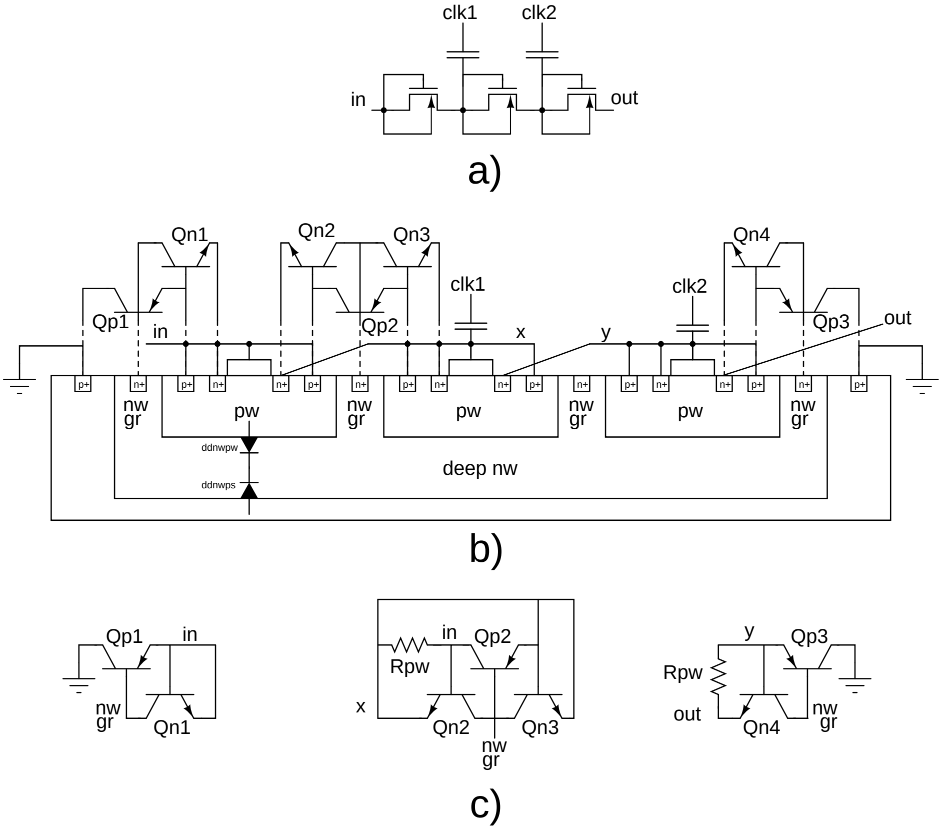


Figure 6.29: a) The schematic of the charge pump (only two stages shown in this illustration). b) The cross-sectional view of the layout to make parasitic BJTs, diodes and resistors visible. c) The simplified view of parasitic BJTs and resistors in different nodes.

	on		on		off		off		off		on		
N#	init.	R1	R2	R1	R2	R1	R2	R1	R2	R1	R2	R1	R2
1	0	4254	3507	523	1080	480	283	0	0	0	0	1061	2255
2	0	3655	3026	1082	545	1830	294	0	0	0	0	276	1076
3	0	0	4517	412	0	0	953	0	0	0	0	309	0
4	0	3388	2151	487	510	214	582	0	0	0	0	1482	3746
5	0	751	682	1366	2888	317	2291	0	0	0	0	2960	985
6	0	680	316	1121	280	2526	1985	0	0	0	0	358	1203
7	0	4450	1758	3327	4547	3406	4391	0	0	0	0	762	834
8	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	295	1978	857	1142	940	1051	1355	1962	0	0	288	2880
10	0	0	287	2040	491	0	0	0	0	0	0	1815	875
11	0	0	0	0	0	0	1390	0	0	0	0	0	3931
12	0	4075	1971	2772	4726	3838	478	0	0	0	0	1372	1387
13	0	3079	3060	2406	717	3536	294	0	0	0	0	459	2168
14	0	0	0	2952	2019	449	507	0	0	0	0	1851	283
15	0	4868	351	3825	4867	2085	2737	0	0	0	0	3038	401
16	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6.1: The registered spike counts for each neuron (N1-N16) in the output layer when the synaptic connections are programmed on and off. Spike counts are read twice (R1 and R2) after each programming cycle. here is no programming in between the read cycles and the input stimulus is always the same. The programming direction (on or off) is indicated in the first row.

Only neurons 8 and 16 are inactive during the whole session, and also neuron 11 has a low activity. As the neuron was biased ultra sensitive (close to self-triggering) it was possible to detect small changes in the synaptic weights. Because of this, the amount of spike counts between respective R1 and R2 varies a lot. The nearly unstable operation also explains why nine read pulses produce thousands of events.

After numerous tries and experiments with different system parameters in order to stabilize the number of generated events while maintaining the programming ability, the programming feature was unexpectedly completely lost. All efforts to restore the programming functionality were unsuccessful despite extensive tests with the available prototype chips; the programming functionality was only observed with one chip for a limited time. Consequently, full-scale learning experiments could not be performed. One possible explanation for the reliability problem is the use of unprotected IO pads, which are sensitive to external disturbances due to the absence of protective circuitry.

Excluding the problems with the programming functionality, the chip responded reasonably to changes in input stimuli and other system parameters, such as bias voltages. For example, by changing the sensitivity of the neuron it was possible to change the ratio of generated output events per the number of input pulses. An example of the registered spike counts in one arbitrarily chosen neuron resulting from an externally given stimulus is presented in table 6.2. In this experiment the test was repeated ten times, the stimulus was given only to one neuron (always the same) and the ap-

trial	1	2	3	4	5	6	7	8	9	10
counts	15	13	12	15	15	13	14	15	16	15

Table 6.2: The registered spike counts for one neuron in the output layer when 500 pulses were applied to one input neuron. Small differences between different trials indicate that the generated spikes result from a stochastic process.

plied input was 500 pulses of width 1 μs and period 1.75 μs per pulse. It can be observed that the registered spike counts are fairly consistent between different trials.

A small variation in registered counts between separate trials reveals the statistical nature of spike generation mechanisms. This is because in addition to accumulated charge due to input pulses, also random noise affects the pulse count: the inability to program the synaptic weights with an arbitrary accuracy combined with a small integrating capacitance causes the neuron to be prone to noise. A (statistically) linear relationship was found between the generated events in the output layer and the number of input pulses. Furthermore, the pulse width and the period of input pulses affected logically to the number of registered events: the shorter the period or the wider the pulses the more counts on the monitored output neuron. In future implementations there are some lessons to learn from the problems in this design. One is that the on-chip voltage distribution must be done carefully when operating close to threshold voltages in time-sensitive circuits. Otherwise, there can be considerable variation in time-constants due to the noise in power supply lines.

The evaluation of the presented prototype chip can be summarized as follows. Taking into account the complexity of the chip, the limited number of internal nodes to be monitored and the fact that the presented chip was the first generation prototype, the results are satisfactory. The chip design was fully functional, but there were some problems in the implementation: the charge pumps performed slightly below target, system noise caused variation to the number of generated events, and the use of unprotected I/O pads caused reliability problems. Despite the problems the functionality of the presented SNN architecture could be demonstrated, since the programming worked for a while and the concept of weight adaptation in an FGT synapse was experimentally demonstrated with another test chip.

Chapter 7

Emerging Analog Computational Elements Beyond CMOS

In this thesis, the focus is on the applicability and challenges of using FGTs as computational elements in analog circuits in deep sub-micron CMOS processes. In the introduction, it was mentioned that the same technology forms the basis for non-volatile digital memory (flash memory). Flash is currently a dominant memory type wherever a significant amount of non-volatile storage is required. A similar commercial breakthrough or "killer application" has not yet been found for the analog counterpart of the FGT. Application of special process options and advances in technology have enabled significant improvements in the read/write times, energy consumption, and capacity of flash memory. There are some possible challengers to flash, for example memristor and Phase Change Memory (PCM) [85]. Both of these devices were invented and their theoretical models were established a long time ago, but feasible implementations of these devices have become possible only due to recent technological advances. Of these two, the memristor will be taken under closer examination since 1) both memristor and PCM are resistance-based memories so that it is sufficient to discuss only one of these two and 2) the energy consumed during the read operation is comparable for the memristor and PCM, but the PCM consumes roughly ten times more energy for the write operation than the memristor [86] (table 2.1 on page 20).

The intrinsic properties of new resistance-based nano-devices do not match the traditional binary data presentation and Boolean logic very well. That is why, the ways for representing and processing data must be reconsidered in order to more efficiently utilize the available properties of the device and to take into account the expected high statistical variability among individual nano-devices. Such variability leads to reduced robustness. The memristor will be briefly examined from this viewpoint. Another interesting perspective is the capability of the memristor to be used as a programmable analog computational element (like FGT). It is discussed whether it is possible to increase the programming resolution of the memristor to the extent which allows a reasonable computational capability, and further discuss the retention of the programmed state.

7.1 Memristor

7.1.1 Definition and Operation Principle

A memristor is a resistor whose resistance value can be changed within a certain restricted range via programming, and ideally, it is able to hold the programmed state until reprogrammed [6]. It is a passive, two-terminal device which provides a functional non-linear relationship between the time integrals of voltage and current. That is, in a memristor the resistance depends on the time integral of the applied input. In the most simplified form, this relationship can be described by the following two equations:

$$v = R(w)i \tag{7.1}$$

$$\frac{dw}{dt} = i \tag{7.2}$$

where R is a generalized resistance that depends on the internal (programmed) state of the device and w is a state variable. With a state variable w representing charge, the two equations above define the behavior of a current-controlled (or charge-controlled) memristor. R has the unit of resistance and is called memristance to differentiate the device from an ordinary resistor. An alternative approach is to consider the memristor as a flux-controlled device, which leads to a concept of memductance with units of conductance [90]. This simple model presented above can be refined to more sophisticated models, such as the one proposed in [91], and described by the following two equations:

$$I = c_1\, w\, sinh(d_1 V(t)) \tag{7.3}$$

$$\frac{dw}{dt} = c_2\, sinh(d_2 V(t)) \tag{7.4}$$

where c_i and d_i, $i = 1,\, 2$ are positive constants, and $w \in [0,\, 1]$ is the state variable of the memristor. This model of memristor includes certain assumptions and boundary conditions, for example the Schottky phenomenon is assumed to be negligible, and the state variable w of the memristor is

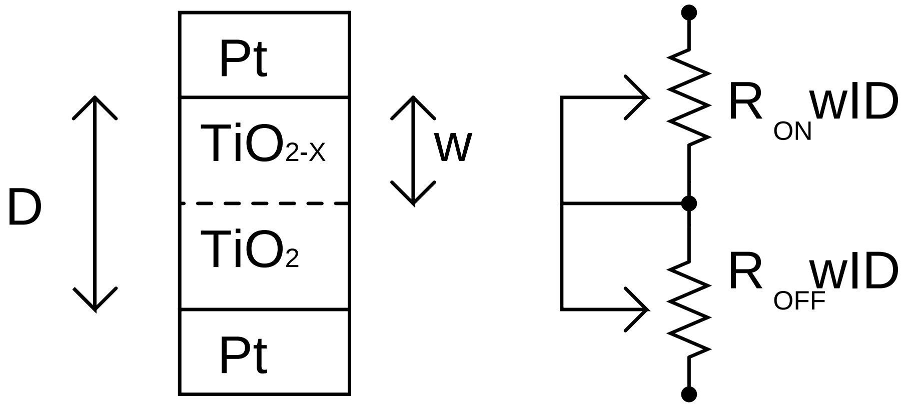


Figure 7.1: The memristor presented by HP labs in [7] comprises a 5 nm thick (D) two layer thin-film sandwiched between two platinum electrodes. One layer is of wideband semiconducting titanium dioxide (TiO_2) with high resistivity and the other of conducting, oxygen-poor (TiO_{2-x}) with low resistivity. The oxygen vacancies move in the applied electric field shifting the weights of series connected variable resistors R_{ON} and R_{OFF}.

assumed to depend highly nonlinearly on the voltage across the memristor. However, due to the focus of this thesis, these advanced models of memristors are not further discussed. In [88], the concept of the memristor was generalized to embody a much broader class of non-linear time-variant systems which they called memristive systems.

The most well-known physical implementation of a memristor is probably the one reported by HP-labs [7]. The memristor created by HP-labs consists of a thin-film of titanium dioxide (TiO2) sandwiched between two platinum electrodes (see Fig. 7.1). Naturally, today there are many other implementations made of different materials, but as the basic mechanism of resistance switching is essentially the same, HP's memristor [90] will be used to briefly review the phenomenon.

According to the prevailing theory, the total resistance of a memristor is composed of a series connection of two variable resistors R_{ON} and R_{OFF}. The application of an external bias across the device creates an electric field in which the oxygen vacancies are able to drift. Consequently, the effective boundary between the two layers moves and the weights of series connected resistors change. Whether the total resistance of the device moves toward a more conductive state or a less conductive state depends on the polarity of the applied field [89].

In practice, the mobility of the ions defines the maximal resistance switching frequency. This frequency can be found by applying an alternating-

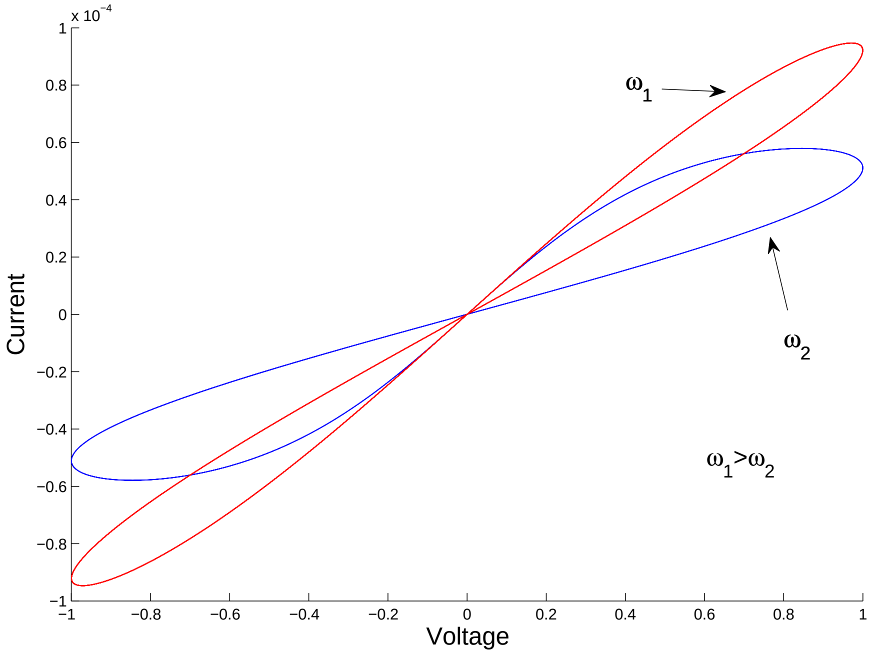


Figure 7.2: The current-voltage response of the memristor shows double-loop hysteresis curve for a sine wave current signal. The hysteresis loop shrinks with the increase in frequency and approaches a straight line. The simulation parameters (matlab model) were chosen arbitrarily due to the illustrative nature of this simulation.

current sinusoidal signal to the memristor. When the frequency is lower than the limiting frequency, the I-V response shows a double-loop hysteresis. For frequencies exceeding the limiting frequency, the hysteresis loop collapses into a straight line and the memristor behaves as an ordinary linear resistor. The characteristic I-V-curve of a memristor is shown in Fig. 7.2. The hysteretic I-V-curve with identical zero-crossing ($v(t)$ and $i(t)$) is a good "fingerprint" when identifying memristive systems.

7.1.2 Implementing functionalities with Memristors - Reflections on the System Level

There are several issues that need to be addressed on the system level when migrating from the accustomed CMOS to passive resistance-based devices. These issues arise from the characteristics of the memristor discussed in the previous section and affect the design on several levels. These issues are discussed in this section in order to highlight some important aspects related to the design.

The memristor is not an energy storage element nor can it provide amplification. With these restrictions in mind it is natural to consider hybrid circuits with some other devices (e.g. CMOS) which are able to supplement the deficiencies of the memristor in order to build complete systems. Memristors can be formed vertically between two intersecting nanowire layers (crossbar), and thus they can be placed above the wiring layers dedicated to interconnecting CMOS without any area penalty. Intuitively, combining resistance switching devices with the flexibility, reliability and high functionality of CMOS seems to be a promising approach. However, if the empty space above the wiring layers is filled with an array of memristors with much higher density than the CMOS devices below, it induces new challenges: How to interface the memristors with the CMOS devices, and how to take into account for the statistical unreliability of nanoscale devices? Because the interfacing and the unreliability problems are common to any application, regardless of whether the memristor is used as a memory or as a computational element, these issues are addressed first.

It has been shown that using two nanowire layers which are slightly rotated in respect to a CMOS array, in order to form a crossbar, the number of memristors can be maximized while maintaining the access to individual devices [82]. This approach both relaxes the density requirements of the interfacing vias and increases redundancy to account for the unreliability of individual devices. The redundancy is of paramount importance so that the desired functionality can be implemented regardless of defective nanodevices, to improve the fault-tolerance of the system.

There are at least two proposed methods on how to form the nanowire crossbar layer on top of CMOS so that these nanoscale devices can be connected to the CMOS devices: CMOS / Molecular Hybrid (CMOL) [82] and Field-Programmable Nanowire Interconnect (FPNI) architecture. A metallic pin (resembling a sharpened via) is used in CMOL approach to connect the nanowire layers to the underlying CMOS, while the connecting pads in FPNI architecture are CMOS scale. As a result, the density of the crossbar is higher with the CMOL architecture [94]. However, these nanopins required by the CMOL architecture present a fabrication challenge, because sharp pins of different heights are needed for the upper and lower nanowire layers. Also, this approach suffers from the uncertainty in the locations of nanopins which can easily result in missing or extra connections. Thus, in FPNI architecture, the density is traded off for easier fabrication and reliability.

Assuming that the interfacing can be implemented reliably with high efficiency, the next challenge arises from the time dependent behavior of memristive devices. Necessary changes on the architectural level originating from this behavior can be demonstrated by considering a digital memory made of memristors. The resistivity of a memristor can be divided into two states: the low resistivity state and high resistivity state and regard it as a binary switch. The binary switch enables non-volatile memory. However, the implementation of a memristor-based non-volatile memory array involves some changes in the way the data is written to or read from the memory. For example, memristor-based memory could be read with an alternating current to prevent changing the memristance value and destroying data. Another possibility for nondestructive readout is to apply only a small voltage across the memristor so that the programming threshold is not exceeded [92]. It should be noted that the change in the state variable is a function of the applied programming voltage and time [91]. The programming threshold is not fixed; it depends on the time scale. In this regard, the programming of the memristor resembles the F-N tunneling of the FGT. An example of a CMOL-based non-volatile binary memory array providing 1 Tbit/cm^2 density is proposed in [93]. When the technology evolves and the yield of functional memristors enhances, improvement up to 100 Tbit/cm^2 can be expected.

In the following, implementing logic computation with memristors is considered as another example of technological challenges in data representation and processing. The memristor does not lend itself very well to Boolean logic due to the inherent dynamical properties. In [96], an alternative for Boolean logic, called material implication, was proposed to be implemented with memristors. The basic operation of material implication is $pIMPq$ (p implies q) which can be described with the truth table 7.1. In table 7.1 p and q are one bit variables, but the same concept can be extended to multi-bit variables. Furthermore, it was shown in [96] that IMP and $False$ operations form a computationally complete logic basis, which means that any operation of Boolean logic can be realized by combining these two operations. For example, Boolean operation $pNANDq$ translates to $pIMP(qIMP0)$ in material implication logic. Memristors are an ideal implementation medium for material implication logic because the same nanoscale switches can be dynamically defined to be either logic gates or memory latches. Thus, the IMP operation can be embedded in the nanoscale memory cells themselves. It has been recently shown that any Boolean function can be implemented with only two memristors allocated for intermediate processing results [97].

7.1.3 Applications to Analog Computation

It has been envisioned that memristors could potentially find use in implementing a digital memory, reconfigurable logic as well as artificial neural networks. All these would be based on a crossbar layer that contains the memristor fabric. The focus of this thesis is on analog computation, so that

p	q	$p \to q$
1	1	1
1	0	0
0	1	1
0	0	1

Table 7.1: Truth table for p implies q (or equivalently $(NOTp)ORq$).

the "digital" applications are beyond the scope of this thesis excluding the examples briefly mentioned in the previous section. Being essentially just a resistor, certainly with valuable programming and memory capability, the memristor or similar nanoscale devices are best suited as an add-on for a future CMOS in large scale, possibly massively parallel, analog applications.

The configuration information needed for the digitally assisted analog circuits discussed earlier in Chapter 1 can be stored in the memristor layer. Existence of dense, low-cost and non-volatile local memory substantially increases the attractiveness of this design scheme. However, it should be noted that especially in analog applications such as providing a switch fabric for a FPAA, the (material dependent) properties of memristors must be selected carefully. For example, if the threshold voltage of an analog memristive switch is too low, the switch acts as a signal dependent resistive element on the signal path and limits the signal to noise ratio. On the other hand, in some analog applications, the same specific characteristic of memristor can be taken advantage of as explained in the next section.

The non-linearity is an inherent property of a memristor and the hysteresis stems from the different thresholds in resistance switching. The non-linearity can be exploited for implementing circuits with CMOS featuring tunable gain. A programmable gain amplifier using CMOS-memristor hybrid presented in [100] is shown in Fig. 7.3. In this example, the differential pulse train is applied to the terminals of a memristor to adjust memristance and ultimately the CMOS amplifier's gain. The advantage over the FGT-based resistor is that a programmable MOS resistor introduces large parasitic capacitances and resistances. However, these kinds of applications (e.g. one amplifier only) do not make the most of the size and density advantage of the nanosize devices. Also, the unreliability of a single device raises a question whether there should be a bank of memristors instead of one device only in case the programmed memristor turned out be defective?

One of the most promising application domains for the memristor is the one which is discussed in Chapter 6 in this thesis: large-scale neuromorphic networks. Within this paradigm, the reconfiguration occurs continuously through adaptation, harnessing the dynamical properties of the memristor.

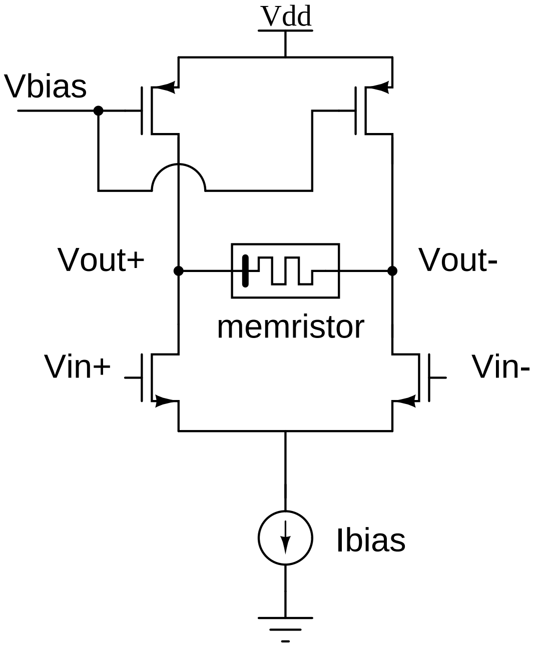


Figure 7.3: The principle of a CMOS differential amplifier featuring a programmable gain [100]. The gain is tuned by a memristor.

Memristors provide the "synapses" for neuromorphic networks, while the CMOS subsystem implements functionally more complex "neurons". The FGT implementation of a synapse with the associated learning mechanism was presented in Chapter 6 in this thesis. The proposed architecture is actually a memristor emulation and thus compatible with all future two-terminal nanodevices that are based on resistivity change.

7.1.4 FGT vs. Memristor

Most of the differences between FGTs and memristors are obvious; a FGT is a four terminal device (transistor), which is capable of providing amplification. Memristor is a passive two terminal device, and thus it can only attenuate signals passing through it. While it is possible to construct multi-input and multi-output FGTs, a memristor always has only two terminals; one input and one output, so that the same terminals have to be used for both reading and programming the state. However, there are also considerable similarities like: Both devices provide a nonlinear I-V-response and an intrinsic memory mechanism which is ideally non-volatile, programming is highly nonlinear (thresholds) and programming mechanisms are based on quantum-mechanical stochastic processes. On the other hand, since gate nodes of transistors can be connected, the state of the FGT can be read

Device	FGT	memristor
type	active	passive
terminals	4+	2
state variable	charge	O ions
non-volatile[1]	yes	yes
state read out	indirect	direct
number of states	continuous	continuous

Table 7.2: Device characteristics comparison. [1] Retention time depends on used materials, material thicknesses and whether the time is measured in power-off or in power-on mode.

and distributed without influencing the state. It is also more difficult to program a two-terminal device in an array without affecting the state of the neighbors instead of a four-terminal FGT device. FGT devices also have a richer repertoire of mechanisms to be used for programming.

Both devices have limited write endurance: after too many memory writes the device eventually breaks down. Write endurance can be meaningfully defined only for a digital memory. According to present knowledge, the memristor can withstand about 100 times more memory writes than an FGT based flash [86] (table 2.1 on page 20). Retention times of both devices change considerably according to the used materials, material thicknesses, whether the retention time is measured in power-off or in power-on mode, temperature and so on. Hence, it is difficult to make a fair comparison based only on the type of the device. On the other hand, a device with a short retention time may match better for a specific application than a device with a long retention time. For example, [98] reports retention loss rate for a memristor which is comparable to short-term memory loss found in biological systems. Perhaps there are memristors with different device properties available in a future device library provided by the foundry, in the same way as there are different transistors (low-voltage, general purpose etc.) in current CMOS device libraries. These would be selected according to the purpose of use and application. The differences and similarities between the memristor and FGT are summarized in table 7.2.

On the other hand, the FGT device can be considered a three-terminal memristive device in certain configurations. For example, the FGT shows a similar hysteretic vi-curve characteristic of a memristive device when an alternating-voltage sinusoidal signal with an appropriate amplitude and frequency is applied to the tunneling junction, and the drain current is regarded as the output. It is also possible to form a tunable resistor from an FGT by controlling the channel resistance with the floating gate charge [99].

Chapter 8

Conclusions

Since the invention of a floating-gate transistor, its usage in both digital and analog applications has been actively researched. So far, commercially the most successful application is the flash memory which provides a non-volatile binary digital memory. There are also many ways to exploit the properties of a FGT as a computational element in analog signal processing or for some other task in analog VLSI design e.g. generating bias voltages/currents, canceling the circuit non-idealities, yield enhancement, as an adaptive element in machine learning etc. Thus, the intrinsic non-volatile memory provided by the FGT plays a key role also in many analog applications.

The nominal gate oxide thickness in deep sub-micron (DSM) process has become too thin to favor charge retention so that the FGTs become leaky if the isolation capability is not improved by special measures. However, it has become increasingly difficult to achieve such process modifications that allow embedding digital flash memory to any SoC, whereas in stand-alone memory circuits the task is easier. It should be noted that the embedded digital flash memory is only available as an IP block and the used memory architecture and related design rules are not directly suitable for analog applications (fixed device size, binary programming etc.). Also, some process options like the consistently used double poly option are not widely available in DSM processes. This hampers scaling down the analog FGTs which in turn prevents the integration benefits and is the main reason why most of the analog circuits utilizing FGTs are typically implemented with the 0.25 μm or older process node. The oxide thickness problem in a scaled down process can be solved by using “double-” or “triple-” oxide transistors intended for providing devices that operate with higher supply voltages than general purpose devices. In practice, the technology scaling poses several challenges which were addressed in this thesis.

In Chapter 2, all possible mechanisms that could be found for programming the non-leaky FGTs in a general purpose CMOS process were reviewed.

In this review, different programming mechanisms were compared on the basis of how many parameters are needed for controlling the gate current, can it be used for adding or removing charge or for both, are there any special needs etc. It was discovered that the frequently used F-N tunneling and IIHEI are the most practical programming mechanisms when a selective (at least in one direction) and accurate bidirectional programming of an array of FGTs is required. Furthermore, in practice the F-N tunneling junction is always needed because electrons can be only added to the floating gate with injection mechanisms and the p-type transistors are more amenable to electron injection (IIHEI). Finally, despite using the p-type transistors for programming the floating gate, the threshold voltage of an n-type transistor can be programmed as well by adopting an indirect programming scheme.

As the migration to DSM process seems not to provide any good alternatives for the F-N tunneling and IIHEI, the feasibility of these methods (in the DSM processes) was discussed in more detail in Chapter 3 and demonstrated with a prototype chip in Chapter 4. It was found out that FGTs are realizable and can be programmed reliably in a scaled down process, but the direct tunneling through the gate oxide can no more be regarded as insignificant and the charge retention capability itself can be compromised even with the thick oxide devices. However, this observation is only suggestive and needs further evidence with e.g. accelerated life time tests. The reflections on performance when replacing ordinary transistors with FGTs were reviewed and estimates as to how the technology scaling affects the performance were drawn. It was discovered that the performance of an FGT device is not severely degraded when compared with a nominal MOS device with similar size in a scaled down process. Furthermore, it is worth remembering that often the flexibility of the FGT can compensate for this loss in performance.

In Chapter 3, low-power subthreshold signal processing was discussed and the possible role of FGTs within this framework was considered. In principle, the properties of FGTs match very well with the requirements of analog signal processing in the subthreshold region but the power efficiency can be easily lost if the programming infrastructure is not carefully designed.

The Chapter 4 was devoted on analyzing the data obtained from several prototype chips. These included IIHEI and F-N tunneling test circuits, an FGT current reference and the autozeroing floating gate amplifier (AFGA). The functionality of IIHEI and FN-tunneling were tested on 180 nm and 130 nm CMOS processes. The current reference was implemented in a 90 nm general purpose CMOS process and the programmability was realized by the F-N tunneling in both directions. The AFGA made of thin oxide devices in a 65 nm CMOS process served as an example of the possibility to take advantage of the gate currents (semi-floating gate techniques) in implementing a desired functionality. Finally a fully functional FGT circuit using a constant charge injection mechanism for programming was presented.

It was discovered that the FGT technology provides an early access method for prototyping computation architectures compatible with all future nanoscale devices that are based on the resistivity manipulation of the material. To this end, an FGT based implementation of a synapse within a context of Spiking Neural Network (SNN) was discussed in Chapter 6.

The proposed synapse implements the STDP learning rule and uses F-N tunneling for weight updates. More specifically, the weight update is based on simultaneously occurring biphasic pulses (the shape of the pulse is specifically designed for this purpose) which superimpose and trigger the F-N tunneling. Furthermore, the temporal aspect of the STDP is realized by the time difference (and the arrival order) of biphasic pulses that determine the strength (and direction) of the weight update. The proposed architecture and learning mechanism is applicable to a memristor to be used as an adaptive element instead of the FGT to implement similar functionality. Moreover, a novel on-chip charge pump capable for providing the biphasic high voltage pulses needed in weight adaptation was also introduced. The novelty of the charge pump lies in its ability to provide pulse shapes with an e-fold increase and decrease around an adjustable DC-level. The amplitude and rise time of the pulse are also tunable.

Although the practical realization of the idea turned out to be more complex than initially estimated, a complete system level implementation of two layer neural network was manufactured in a 65 nm CMOS in order to proof the concept. The per neuron charge pumps did not work exactly as designed and consequently the synaptic weights could not be systematically changed. However, on one test chip, the synapses were successfully programmed on and off several times in a row. Excluding the slightly unsuccessful charge pump design, all sub-blocks in this relatively complex prototype chip seemed to be fully functional. The neurons in the output layer were able to generate spikes when stimulating the neurons in the input layer. In addition, the number of observed events in the output behaved in accordance with expectations in respect to all system parameters. Nevertheless, the concept of weight adaptation in an FGT synapse was experimentally demonstrated with another test chip.

The memristor was taken into closer examination in Chapter 7 due to many common characteristics with FGTs. The operation principle of the memristor was reviewed and possible applications were considered. Finally, a small scale comparison between memristor and FGT was made.

Bibliography

[1] http://www.ti.com/corp/docs/press/backgrounder/analog.shtml

[2] A. Basu, S. Brink, C. Schlottmann, S. Ramakrishnan, C. Petre, S. Koziol, F. Baskaya, C.M. Twigg, P. Hasler 'A Floating-Gate-Based Field-Programmable Analog Array', IEEE Journal of Solid-State Circuits, Vol.45 , No. 9, pp. 1781-1794, 2010.

[3] M. Pankaala, K. Virtanen, A. Paasio, 'An Analog 2-D DCT Processor', IEEE Transactions on Circuits and Systems for Video Technology, Vol. 16 , No. 10, pp. 1209-1216, 2006.

[4] M. Pankaala, 'On Designing an Analog Current-Mode 2-D Discrete Cosine Transform Processor', Licentiate's thesis, University of Turku, Department of Information Technology, Microelectronics, 2007.

[5] L. Fang-shi, L. Yung-Fu, A. Weng, K. Hsueh, H. Fu-Lung, 'Digitally-assisted analog designs for submicron CMOS technology', Proceedings of 2010 International Symposium on VLSI Design Automation and Test, pp. 49-52, 2010.

[6] Leon O. Chua, 'Memristor - The Missing Circuit Element', IEEE Transactions on Circuit Theory, Vol. CT-18, No. 5, pp. 507-519, 1971.

[7] D. Strukov, et al., 'The missing memristor found', Nature, Vol. 453, pp. 80-83, May 2008.

[8] Woo Yeong Cho, et al., 'A 0.18-um 3.0-V 64-Mb nonvolatile phase-transition random access memory (PRAM)', IEEE Journal of Solid-State Circuits, Vol.40 , No. 1, pp. 293-300, 2005.

[9] M. Durlam, et al., 'A 1-Mbit MRAM based on 1T1MTJ bit cell integrated with copper interconnects', IEEE Journal of Solid-State Circuits, Vol. 38 , No. 5, pp. 769-773, 2003.

[10] K. Rahimi, C. Diorio, C. Hernandez, M.D. Brockhausen, 'A Simulation Model for Floating-gate MOS Synapse Transistors', IEEE International Symposium on Circuits and Systems, Vol. 2, pp. II-532-II-535, 2002.

[11] A-J Annema, B. Nauta, R. van Langevelde, H. Tuinhout 'Analog Circuits in Ultra- Deep-submicron CMOS', IEEE, International Journal of Solid-State Circuits, Vol. 40, No.1, pp. 132-143, 2005.

[12] D. Kahng, S.M. Sze, 'A Floating-gate and its applications to memory devices', The Bell System Technical Journal, XLVI(6), pp. 1288-1295, 1967.

[13] B.A. Minch, P. Hasler, C. Diorio, 'Multiple-input translinear element networks', IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, Vol. 48, No. 1, pp. 20-28, 2001.

[14] A. Sil, 'Hot Carrier Effects in Deep Submicron CMOS', http://www.cacs.louisiana.edu/labs/vlsi_old/secure/presentations/FALL05/abhijit-hot-carrier-effect.ppt, 2005.

[15] D.W. Graham, E. Farquhar, B. Degnan, C. Gordon, P. Hasler, 'Indirect Programming of Floating-Gate Transistors', IEEE Transactions on Circuits and Systems I: Regular Papers, Vol. 54 , No. 5, pp. 951-963, 2007.

[16] M.N. Martin, D.R. Roth, A. Garrison-Darrin, P.J. McNulty, A.G. Andreou, ' FGMOS dosimetry: design and implementation', IEEE Transactions on Nuclear Science, Vol. 48 , No. 6, pp. I 2050-2055, 2001.

[17] P. Hasler, B.A Minch, C. Diorio, 'An autozeroing floating-gate amplifier', IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, Vol. 48 , No. 1, pp. 74-82, 2001.

[18] A. Bandyopadhyay, Lee Jungwon, R.W. Robucci, P. Hasler, 'MATIA: a programmable 80 μW/frame CMOS block matrix transform imager architecture', IEEE Journal of Solid-State Circuits, Vol. 41, No. 3, pp. 663-672, 2006.

[19] M.I. Vexler, S.E. Tyaginov, A.F. Shulekin, 'Determination of the hole effective mass in thin silicon dioxide film by means of an analysis of characteristics of a MOS tunnel emitter transistor', Journal of Physics: Condensed Matter, Vol. 17, No. 50, 2005.

[20] B. Prince, 'Semiconductor Memories', John Wiley & Sons Ltd., ISBN 0 471 92465 2, 1983.

[21] T. Guoqiao, H. Chauveau, D. Boter, D. Dormans, R. Verhaar, 'A simple and accurate method to extract neutral threshold voltage of floating gate flash devices and its application to flash reliability characterization', IEEE International Integrated Reliability Workshop Final Report, pp. 52-56, 2007.

[22] C. Chang, M-S. Liang, C.Hu, R.W. Brodersen, 'Carrier Tunneling Related Phenomena In Thin Oxide MOSFETs', International Electron Devices Meeting, Vol. 29, pp. 194-197, 1983.

[23] W.K. Chim, 'Semiconductor Device and Failure Analysis', John Wiley & Sons Ltd., ISBN 0 471 49240 X, 2000.

[24] F. Henrici, C. Peters, J. Becker, M. Ortmanns, Y. Manoli, 'Reliability Study of Single-Poly Floating Gates in 0.13 um CMOS for use in Field Programmable Analog Arrays', Midwest Symposium on Circuits and Systems, pp. 17-20, 2008.

[25] I.St. John, R.M. Fox, 'Leakage effects in metal-connected floating-gate circuits', IEEE Transactions on Circuits and Systems II: Express Briefs, Vol. 53 , No. 7, pp. 577-579, 2006.

[26] E. Sackinger, W. Guggenbuhl, An Analog Trimming Circuit Based on a Floating-Gate Device', IEEE Journal of Solid-State Circuits, Vol. 23, No.6, pp.1437-1440 ,1998.

[27] M. Lenzlinger, E.H. Snow, 'Fowler-Nordheim tunneling into thermally grown SiO2', Journal of Applied Physics, Vol. 40, No. 6, pp. 278-283, 1969.

[28] E. Miranda, G. Redin, A. Faign, 'An effective-field approach for the Fowler-Nordheim tunneling current through a metal-oxide', Journal of Applied Physics, Vol. 82, No. 3, pp. 1262-1265, 1997.

[29] L.R. Carley, 'Trimming analog circuits using floating-gate analog MOS memory', IEEE Journal of Solid-State Circuits, Vol. 24 , No. 6, pp. 1569-1575, 1989.

[30] W. Wen, D. Xiaodong, J.S. Yuan, 'Modeling of time-dependent dielectric breakdown in copper metallization', IEEE Transactions on Device and Materials Reliability, Vol. 3, No. 2 pp. 26-30, 2003.

[31] K. Ohsaki, N. Asamoto, S. Takagaki, 'A single poly EEPROM cell structure for use in standard CMOS processes', IEEE Journal of Solid-State Circuits, Vol. 29 , No. 3, pp. 311-316, 1994.

[32] A.J.Annema, G.G.Geelen, P. de Jong, '5.5V Tolerant I/O in a 2.5V 0.25um CMOS Technology', in Proc. IEEE Custom Integrated Circuit Conference, pp. 417-420, 2000.

[33] D.K Schroder, 'Semiconductor material and device characterization', John Wiley & Sons Ltd., ISBN 0 471 24139 3, 1998.

[34] J.J. Sanchez, T.A. DeMassa, 'Review of carrier injection in the silicon/silicon-dioxide system', IEEE Proceedings, Vol. 138, No. 3, 1991.

[35] N-C. Peng et al., 'Single-poly EEPROM', United States Patent Application, No. US20060208306, 2006.

[36] P. Hasler, 'Foundations of Learning in Analog VLSI', Ph.D thesis, Department of Computation and Neural Systems, California Institute of Technology, 1997.

[37] C. Duffy, P.Hasler, 'Modeling Hot-Electron Injection in pFET's', Journal of Computational Electronics, Vol. 2, No.2-4, pp. 317-322, 2003.

[38] G.J. Serrano, 'High Performance Analog Circuit Design Using Floating-gate Techniques', Ph.D thesis, School of Electrical and Computer Engineering, Georgia Institute of Technology, 2007.

[39] G.J. Serrano, P. Hasler, 'A Precision Low-TC Wide-Range CMOS Current Reference', IEEE Journal of Solid-State Circuits, Vol. 43 , No. 2, pp. 558-565, 2008.

[40] H. Nozama, S. Kokyama, 'A thermionic electron emission model for charge retention in SAMOS structures', Japanese Journal of Applied Physics, vol. 21, pp. L111-L112, Feb. 1992.

[41] L. Huang, M. Ashouei, F. Yazicioglu, J. Penders, R. Vullers, G. Dolmans, 'Ultra-Low Power Sensor Design for Wireless Body Area Networks: Challenges, Potential Solutions, and Applications', JDCTA: International Journal of Digital Content Technology and its Applications, Vol. 3, No. 3, pp. 136-148, 2009.

[42] R. Sarpeshkar, 'Ultra low power bioelectronics: fundamentals, biomedical applications, and bio-inspired systems', Cambridge University Press, ISBN 978-0-521-85727-7, 2010.

[43] N. Lajnef, S. Chakrabartty, N. Elvin, A. Elvin, 'A sub-mircowatt piezo-floating-gate sensor for long-term fatique monitoring in biomechanical implants', Proceedings of the 28th IEEE EMBS Annual International Conference, pp. 5936-5939, 2006.

[44] P. Hasler, 'Low-Power Programmable Signal Processing', Proceedings of the Fifth International Workshop on System-on-Chip for Real-Time Applications, pp. 413-418, 2005.

[45] J. Maunu, M. Pankaala, J. Marku, J. Poikonen, M. Laiho, A. Paasio, 'Current source calibration by combination selection of minimum sized devices', IEEE International Symposium on Circuits and Systems, pp. 549-552, 2006.

[46] K. Rahimi, C. Diorio, C. Hernandez, M.D. Brockhausen, 'A simulation model for floating-gate MOS synapse transistors', IEEE International Symposium on Circuits and Systems, Vol. 2, pp. 532-535, 2002.

[47] M. Pankaala, J. Maunu, M. Laiho, A. Paasio, 'Experiments with Floating Gate Devices in 0.18 and 0.13 Micron CMOS Technologies' IEEE Norchip Conference, pp. 59-62, 2006.

[48] J. Hyde, T. Humes, C. Diorio, M. Thomas, M, M.'A 300-MS/s 14-bit digital-to-analog converter in logic CMOS' IEEE Journal of Solid-State Circuits, Vol. 38 , No. 5, pp. 734-740, 2003.

[49] C.A. Mead, 'Scaling of MOS technology to submicrometer feature sizes', The Journal of VLSI Signal Processing, Vol. 8, No. 1, pp. 9-25, 1994.

[50] F. Lara-Villa, F. Yanez-Ortega, A.L Mota-Rodriguez, I. Padilla-Cantoya, A. Diaz-Sanchez, J.M Rocha-Perez, J.E. Molinar-Solis, 'A novel divider using the Gilbert's cell with Floating Gate feedback', IEEE International Midwest Symposium on Circuits and Systems, pp. 160-163, 2009.

[51] R. Chawla, 'Power-Efficient Analog Systems to Perform Signal-Processing Using Floating-Gate MOS Device for Portable Applications', Ph.D thesis, School of Electrical and Computer Engineering, Georgia Institute of Technology, 2004.

[52] http://www.arm.com/products/processors/cortex-m/cortex-m0.php

[53] The Blue Brain Project, http://bluebrain.epfl.ch/

[54] E.D. Farquhar, 'Summary and Impact of Large Scale Field-Programmable Analog Neuron Arrays (FPNAs)', Ph.D thesis, School of Electrical and Computer Engineering, Georgia Institute of Technology, 2005.

[55] http://www.stanford.edu/group/brainsinsilicon/index.html

[56] http://brainscales.kip.uni-heidelberg.de

[57] C.A. Mead, 'Analog VLSI and Neural Systems', Addison Wesley Publishing Company, ISBN-10: 0201059924, 1989.

[58] J. Vreeken, 'Spiking neural networks, an introduction', Department of Information and Computing Sciences, Utrecht University, Tech. Rep. UU-CS-2003-008, 2003.

[59] E.M. Izhikevich, 'Dynamical Systems in Neuroscience', The MIT Press Cambridge, Massachusetts London, England, ISBN 978-0-262-09043-8, 2007.

[60] E.M. Izhikevich, 'Which Model to Use for Cortical Spiking Neurons', IEEE Transactions on Neural Networks, Vol. 15, No. 5, pp. 1063-1070, 2004.

[61] R. Serrano-Gotarredona et al., 'CAVIAR: A 45k-Neuron, 5M-Synapse, 12G-connects/sec AER Hardware Sensory-Processing-Learning-Actuating System for High Speed Visual Object Recognition and Tracking', IEEE Transactions on Neural Networks, vol. 20, No. 9, pp. 1417-1438, 2009.

[62] J. Bose, 'Engineering a Sequence Machine Through Spiking Neurons Employing Rank-order Codes', Ph.D thesis, School of Computer Science, University of Manchester, 2007.

[63] R.B. Wells, 'Cortical Neurons and Circuits: A Tutorial Introduction', http://www.mrc.uidaho.edu/ rwells/techdocs/Cortical %20Neurons%20and%20Circuits.pdf

[64] L. F. Abbott, 'Lapicque's introduction of the integrate-and-fire model neuron (1907)', Brain Research Bulletin, Vol. 50, Nos. 5/6, pp. 303-304, 1999.

[65] B.W. Knight, 'Dynamics of Encoding in a Population of Neurons', Journal of General Physiology, Vol. 59, No. 6, pp. 734-766, 1972.

[66] A. Hodgkin, A. Huxley, 'A quantitative description of membrane current and its application to conduction and excitation in nerve', Journal of Physiology, Vol. 117, No. 4, pp. 500-544, 1952.

[67] T. Heimburg, A.D. Jackson, 'On soliton propagation in biomembranes and nerves', Proceedings of the National Academy of Sciences (PNAS), Vol. 102 No. 28, pp. 9790-9795, 2005.

[68] E.M. Izhikevich, 'Which model to use for cortical spiking neurons?', Neural Networks, Vol. 15, No. 5, pp. 1063-1070, 2004.

[69] J.H.B. Wijekoon and P. Dudek, 'Compact silicon neuron circuit with spiking and bursting behaviour', Neural Networks, Vol. 21, No. 2-3, pp. 524-534, 2008.

[70] R. Serrano-Gotarredona et al., 'AER Building Blocks for Multi-Layer Multi-Chip Neuromorphic Vision Systems', Advances in Neural Information Processing Systems, Vol. 18, pp. 1217-1224, 2005.

[71] J. Xin, M. Lujan, L:A: Plana, S. Davies, S. Temple, S.B. Furber, 'Modeling Spiking Neural Networks on spiNNaker', Computing in Science & Engineering, Vol. 12, No. 5, pp. 91-97, 2010.

[72] J. Schemmel, D. Brüderle, A. Grübl, M. Hock, K. Meier, S. Millner, 'A Wafer-Scale Neuromorphic Hardware System for Large-Scale Neural Modeling', Proceedings of 2010 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 1947-1950, 2010.

[73] D.O. Hebb, 'The Organization of Behavior: A Neuropsychological Theory', John Wiley & Sons Ltd., ISBN 0-8058-4300-0, 1949.

[74] S. Song, K.D. Miller, L.F. Abbott, 'Competitive Hebbian learning through spike-timing-dependent-plasticity', Nature Neuroscience, Vol. 3, No. 9, pp. 919-926, 2000.

[75] B. Linares-Barranco, T. Serrano-Gotarredona, 'Memristance can explain Spike-Time- Dependent-Plasticity in Neural Synapses', Nature Precedings, 2009.

[76] G.S. Snider, 'Spike-Timing-Dependent Learning in Memristive Nanodevices', IEEE International Symposium on Nanoscale Architectures, pp. 85-92, 2008.

[77] G.S. Snider and R.S. Williams, 'Nano/CMOS architectures using a field-programmable nanowire interconnect', Nanotechnology, Vol. 18, No. 3, 2007.

[78] M. Pankaala, M. Laiho, P. Hasler, 'Compact Floating-gate Learning Array with STDP', International Joint Conference on Neural Networks, pp. 2409-2415, 2009.

[79] A.M. Haas, 'Compact Circuits and Adaptation Techniques for Implementing Adaptive Neurons and Synapses with Spike Timing Dependent Plasticity (STDP)', Patent application number US20090292661, 2009.

[80] A. Afifi, A. Ayatollahi, F. Raissi, 'Implementation of biologically plausible spiking neural network models on the memristor crossbar-based CMOS/nano circuits', European Conference on Circuit Theory and Design, pp. 563-566, 2009.

[81] A. Afifi, A. Ayatollahi, F. Raissi, 'CMOL implementation of spiking neurons and spike-timing dependent plasticity', International Journal of Circuit Theory and Applications, 2010.

[82] Ö. Türel, J.H. Lee, X. Ma and K. Likharev, 'Neuromorphic architectures for nanoelectronic circuits', International Journal of Circuit Theory and Applications, Vol. 32, No. 5, pp. 277-302, 2004.

[83] P.D. Smith, P. Hasler, 'A Programmable Diffuser Circuit Based on Floating-gate Devices', Midwest Symposium on Circuits and Systems, Vol. 1, No. 1, pp. 291-294, 2002.

[84] P. Lichtsteiner, C. Posch and T. Delbrück, 'A 128 X 128 120db 30mw asynchronous vision sensor that responds to relative intensity change', IEEE International Solid-State Circuits Conference, Digest of Technical Papers, pp. 2060-2069, 2006.

[85] http://www.zurich.ibm.com/news/11/pcm.html

[86] D.A. Roberts, 'Efficient Data Center Architectures Using Non-Volatile Memory and Reliability Techniques', Ph.D thesis, Department of Computer Science and Engineering, University of Michigan, 2011.

[87] B. Widrow, 'An Adaptive "Adaline" Neuron Using Chemical "Memistors"', Stanford Electronics Laboratories Technical Report 1553-2, October 1960.

[88] L.O. Chua, S.M. Kang, 'Memristive devices and systems', Proceedings of IEEE, Vol. 64, No. 2, pp.209-223, 1976.

[89] D. B. Strukov et al., 'Hybrid CMOS/Memristor Circuits', Proceedings of 2010 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 1967-1970, 2010.

[90] K. Kerur, 'A Study of The Memristor, the Fourth Circuit Element', An M.Sc. Thesis, Kansas State University, Manhattan, Kansas, 2010.

[91] E. Lehtonen, J. Poikonen, M. Laiho, W. Lu, 'Time-dependency of the threshold voltage in memristive devices', Proceedings of 2011 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 2245-2248, 2011.

[92] M. Laiho, E. Lehtonen, 'Cellular nanoscale network cell with memristors for local implication logic and synapses', Proceedings of 2010 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 2051-2054, 2010.

[93] D. B. Strukov and K. K. Likharev, 'Defect-tolerant architectures for nanoelectronic crossbar memories', Journal of Nanoscience and Nanotechnology, Vol. 7, pp. 151-167, 2007.

[94] G. Snider, R.S.Williams, 'Nano/CMOS architecture using a field-programmable nanowire interconnect', Nanotechnology, Vol. 18, No. 3, 2007.

[95] D. B. Strukov and K. K. Likharev, 'CMOL FPGA: A cell-based, reconfigurable architecture for hybrid digital circuits using twoterminal nanodevices', Nanotechnology, Vol. 16, pp. 888-900, 2005.

[96] J. Borghetti1, G. Snider, P. Kuekes, J. Yang, D. Stewart, S. Williams,'Memristive switches enable stateful logic operations via material implication', Nature letter, Vol. 464, pp. 873-876, 2010.

[97] E. Lehtonen, J.H. Poikonen, M. Laiho, 'Two Memristors Suffice to Compute All Boolean Functions', Electronics Letters, Vol. 46, No. 3, pp.239-240, February 2010.

[98] T. Chang, S-H. Jo and W. Lu, 'Short-Term Memory to Long-Term Memory Transition in a Nanoscale Memristor', ACS Nano, Vol. 5, No. 9, pp. 7669-7676, 2011.

[99] E. Ozalevli, P.E. Hasler, 'A tunable floating gate CMOS resistor for low-power and low-voltage applications', IEEE International Symposium on Circuits and Systems, pp. 4273-4276, 2006.

[100] Sangho Shin, Kyungmin Kim, Sung-Mo Kang, 'Memristor Applications for Programmable Analog ICs', IEEE Transactions on Nanotechnology, Vol. 10, No. 2, pp. 266-274, 2011.

Turku Centre for Computer Science
TUCS Dissertations

1. **Marjo Lipponen**, On Primitive Solutions of the Post Correspondence Problem
2. **Timo Käkölä**, Dual Information Systems in Hyperknowledge Organizations
3. **Ville Leppänen**, Studies on the Realization of PRAM
4. **Cunsheng Ding**, Cryptographic Counter Generators
5. **Sami Viitanen**, Some New Global Optimization Algorithms
6. **Tapio Salakoski**, Representative Classification of Protein Structures
7. **Thomas Långbacka**, An Interactive Environment Supporting the Development of Formally Correct Programs
8. **Thomas Finne**, A Decision Support System for Improving Information Security
9. **Valeria Mihalache**, Cooperation, Communication, Control. Investigations on Grammar Systems.
10. **Marina Waldén**, Formal Reasoning About Distributed Algorithms
11. **Tero Laihonen**, Estimates on the Covering Radius When the Dual Distance is Known
12. **Lucian Ilie**, Decision Problems on Orders of Words
13. **Jukkapekka Hekanaho**, An Evolutionary Approach to Concept Learning
14. **Jouni Järvinen**, Knowledge Representation and Rough Sets
15. **Tomi Pasanen**, In-Place Algorithms for Sorting Problems
16. **Mika Johnsson**, Operational and Tactical Level Optimization in Printed Circuit Board Assembly
17. **Mats Aspnäs**, Multiprocessor Architecture and Programming: The Hathi-2 System
18. **Anna Mikhajlova**, Ensuring Correctness of Object and Component Systems
19. **Vesa Torvinen**, Construction and Evaluation of the Labour Game Method
20. **Jorma Boberg**, Cluster Analysis. A Mathematical Approach with Applications to Protein Structures
21. **Leonid Mikhajlov**, Software Reuse Mechanisms and Techniques: Safety Versus Flexibility
22. **Timo Kaukoranta**, Iterative and Hierarchical Methods for Codebook Generation in Vector Quantization
23. **Gábor Magyar**, On Solution Approaches for Some Industrially Motivated Combinatorial Optimization Problems
24. **Linas Laibinis**, Mechanised Formal Reasoning About Modular Programs
25. **Shuhua Liu**, Improving Executive Support in Strategic Scanning with Software Agent Systems
26. **Jaakko Järvi**, New Techniques in Generic Programming – C++ is more Intentional than Intended
27. **Jan-Christian Lehtinen**, Reproducing Kernel Splines in the Analysis of Medical Data
28. **Martin Büchi**, Safe Language Mechanisms for Modularization and Concurrency
29. **Elena Troubitsyna**, Stepwise Development of Dependable Systems
30. **Janne Näppi**, Computer-Assisted Diagnosis of Breast Calcifications
31. **Jianming Liang**, Dynamic Chest Images Analysis
32. **Tiberiu Seceleanu**, Systematic Design of Synchronous Digital Circuits
33. **Tero Aittokallio**, Characterization and Modelling of the Cardiorespiratory System in Sleep-Disordered Breathing
34. **Ivan Porres**, Modeling and Analyzing Software Behavior in UML
35. **Mauno Rönkkö**, Stepwise Development of Hybrid Systems
36. **Jouni Smed**, Production Planning in Printed Circuit Board Assembly
37. **Vesa Halava**, The Post Correspondence Problem for Market Morphisms
38. **Ion Petre**, Commutation Problems on Sets of Words and Formal Power Series
39. **Vladimir Kvassov**, Information Technology and the Productivity of Managerial Work
40. **Frank Tétard**, Managers, Fragmentation of Working Time, and Information Systems

41. **Jan Manuch**, Defect Theorems and Infinite Words
42. **Kalle Ranto**, Z_4-Goethals Codes, Decoding and Designs
43. **Arto Lepistö**, On Relations Between Local and Global Periodicity
44. **Mika Hirvensalo**, Studies on Boolean Functions Related to Quantum Computing
45. **Pentti Virtanen**, Measuring and Improving Component-Based Software Development
46. **Adekunle Okunoye**, Knowledge Management and Global Diversity – A Framework to Support Organisations in Developing Countries
47. **Antonina Kloptchenko**, Text Mining Based on the Prototype Matching Method
48. **Juha Kivijärvi**, Optimization Methods for Clustering
49. **Rimvydas Rukšėnas**, Formal Development of Concurrent Components
50. **Dirk Nowotka**, Periodicity and Unbordered Factors of Words
51. **Attila Gyenesei**, Discovering Frequent Fuzzy Patterns in Relations of Quantitative Attributes
52. **Petteri Kaitovaara**, Packaging of IT Services – Conceptual and Empirical Studies
53. **Petri Rosendahl**, Niho Type Cross-Correlation Functions and Related Equations
54. **Péter Majlender**, A Normative Approach to Possibility Theory and Soft Decision Support
55. **Seppo Virtanen**, A Framework for Rapid Design and Evaluation of Protocol Processors
56. **Tomas Eklund**, The Self-Organizing Map in Financial Benchmarking
57. **Mikael Collan**, Giga-Investments: Modelling the Valuation of Very Large Industrial Real Investments
58. **Dag Björklund**, A Kernel Language for Unified Code Synthesis
59. **Shengnan Han**, Understanding User Adoption of Mobile Technology: Focusing on Physicians in Finland
60. **Irina Georgescu**, Rational Choice and Revealed Preference: A Fuzzy Approach
61. **Ping Yan**, Limit Cycles for Generalized Liénard-Type and Lotka-Volterra Systems
62. **Joonas Lehtinen**, Coding of Wavelet-Transformed Images
63. **Tommi Meskanen,** On the NTRU Cryptosystem
64. **Saeed Salehi**, Varieties of Tree Languages
65. **Jukka Arvo**, Efficient Algorithms for Hardware-Accelerated Shadow Computation
66. **Mika Hirvikorpi**, On the Tactical Level Production Planning in Flexible Manufacturing Systems
67. **Adrian Costea**, Computational Intelligence Methods for Quantitative Data Mining
68. **Cristina Seceleanu**, A Methodology for Constructing Correct Reactive Systems
69. **Luigia Petre**, Modeling with Action Systems
70. **Lu Yan**, Systematic Design of Ubiquitous Systems
71. **Mehran Gomari**, On the Generalization Ability of Bayesian Neural Networks
72. **Ville Harkke**, Knowledge Freedom for Medical Professionals – An Evaluation Study of a Mobile Information System for Physicians in Finland
73. **Marius Cosmin Codrea**, Pattern Analysis of Chlorophyll Fluorescence Signals
74. **Aiying Rong**, Cogeneration Planning Under the Deregulated Power Market and Emissions Trading Scheme
75. **Chihab BenMoussa**, Supporting the Sales Force through Mobile Information and Communication Technologies: Focusing on the Pharmaceutical Sales Force
76. **Jussi Salmi**, Improving Data Analysis in Proteomics
77. **Orieta Celiku**, Mechanized Reasoning for Dually-Nondeterministic and Probabilistic Programs
78. **Kaj-Mikael Björk**, Supply Chain Efficiency with Some Forest Industry Improvements
79. **Viorel Preoteasa**, Program Variables – The Core of Mechanical Reasoning about Imperative Programs
80. **Jonne Poikonen**, Absolute Value Extraction and Order Statistic Filtering for a Mixed-Mode Array Image Processor
81. **Luka Milovanov**, Agile Software Development in an Academic Environment
82. **Francisco Augusto Alcaraz Garcia**, Real Options, Default Risk and Soft Applications
83. **Kai K. Kimppa**, Problems with the Justification of Intellectual Property Rights in Relation to Software and Other Digitally Distributable Media
84. **Dragoş Truşcan**, Model Driven Development of Programmable Architectures
85. **Eugen Czeizler**, The Inverse Neighborhood Problem and Applications of Welch Sets in Automata Theory

86. **Sanna Ranto**, Identifying and Locating-Dominating Codes in Binary Hamming Spaces
87. **Tuomas Hakkarainen**, On the Computation of the Class Numbers of Real Abelian Fields
88. **Elena Czeizler**, Intricacies of Word Equations
89. **Marcus Alanen**, A Metamodeling Framework for Software Engineering
90. **Filip Ginter**, Towards Information Extraction in the Biomedical Domain: Methods and Resources
91. **Jarkko Paavola**, Signature Ensembles and Receiver Structures for Oversaturated Synchronous DS-CDMA Systems
92. **Arho Virkki**, The Human Respiratory System: Modelling, Analysis and Control
93. **Olli Luoma**, Efficient Methods for Storing and Querying XML Data with Relational Databases
94. **Dubravka Ilić**, Formal Reasoning about Dependability in Model-Driven Development
95. **Kim Solin**, Abstract Algebra of Program Refinement
96. **Tomi Westerlund**, Time Aware Modelling and Analysis of Systems-on-Chip
97. **Kalle Saari**, On the Frequency and Periodicity of Infinite Words
98. **Tomi Kärki**, Similarity Relations on Words: Relational Codes and Periods
99. **Markus M. Mäkelä**, Essays on Software Product Development: A Strategic Management Viewpoint
100. **Roope Vehkalahti**, Class Field Theoretic Methods in the Design of Lattice Signal Constellations
101. **Anne-Maria Ernvall-Hytönen**, On Short Exponential Sums Involving Fourier Coefficients of Holomorphic Cusp Forms
102. **Chang Li**, Parallelism and Complexity in Gene Assembly
103. **Tapio Pahikkala**, New Kernel Functions and Learning Methods for Text and Data Mining
104. **Denis Shestakov**, Search Interfaces on the Web: Querying and Characterizing
105. **Sampo Pyysalo**, A Dependency Parsing Approach to Biomedical Text Mining
106. **Anna Sell**, Mobile Digital Calendars in Knowledge Work
107. **Dorina Marghescu**, Evaluating Multidimensional Visualization Techniques in Data Mining Tasks
108. **Tero Säntti**, A Co-Processor Approach for Efficient Java Execution in Embedded Systems
109. **Kari Salonen**, Setup Optimization in High-Mix Surface Mount PCB Assembly
110. **Pontus Boström**, Formal Design and Verification of Systems Using Domain-Specific Languages
111. **Camilla J. Hollanti**, Order-Theoretic Mehtods for Space-Time Coding: Symmetric and Asymmetric Designs
112. **Heidi Himmanen**, On Transmission System Design for Wireless Broadcasting
113. **Sébastien Lafond**, Simulation of Embedded Systems for Energy Consumption Estimation
114. **Evgeni Tsivtsivadze**, Learning Preferences with Kernel-Based Methods
115. **Petri Salmela**, On Commutation and Conjugacy of Rational Languages and the Fixed Point Method
116. **Siamak Taati**, Conservation Laws in Cellular Automata
117. **Vladimir Rogojin**, Gene Assembly in Stichotrichous Ciliates: Elementary Operations, Parallelism and Computation
118. **Alexey Dudkov**, Chip and Signature Interleaving in DS CDMA Systems
119. **Janne Savela**, Role of Selected Spectral Attributes in the Perception of Synthetic Vowels
120. **Kristian Nybom**, Low-Density Parity-Check Codes for Wireless Datacast Networks
121. **Johanna Tuominen**, Formal Power Analysis of Systems-on-Chip
122. **Teijo Lehtonen**, On Fault Tolerance Methods for Networks-on-Chip
123. **Eeva Suvitie**, On Inner Products Involving Holomorphic Cusp Forms and Maass Forms
124. **Linda Mannila**, Teaching Mathematics and Programming – New Approaches with Empirical Evaluation
125. **Hanna Suominen**, Machine Learning and Clinical Text: Supporting Health Information Flow
126. **Tuomo Saarni**, Segmental Durations of Speech
127. **Johannes Eriksson**, Tool-Supported Invariant-Based Programming

128. **Tero Jokela**, Design and Analysis of Forward Error Control Coding and Signaling for Guaranteeing QoS in Wireless Broadcast Systems
129. **Ville Lukkarila**, On Undecidable Dynamical Properties of Reversible One-Dimensional Cellular Automata
130. **Qaisar Ahmad Malik**, Combining Model-Based Testing and Stepwise Formal Development
131. **Mikko-Jussi Laakso**, Promoting Programming Learning: Engagement, Automatic Assessment with Immediate Feedback in Visualizations
132. **Riikka Vuokko**, A Practice Perspective on Organizational Implementation of Information Technology
133. **Jeanette Heidenberg**, Towards Increased Productivity and Quality in Software Development Using Agile, Lean and Collaborative Approaches
134. **Yong Liu**, Solving the Puzzle of Mobile Learning Adoption
135. **Stina Ojala**, Towards an Integrative Information Society: Studies on Individuality in Speech and Sign
136. **Matteo Brunelli**, Some Advances in Mathematical Models for Preference Relations
137. **Ville Junnila**, On Identifying and Locating-Dominating Codes
138. **Andrzej Mizera**, Methods for Construction and Analysis of Computational Models in Systems Biology. Applications to the Modelling of the Heat Shock Response and the Self-Assembly of Intermediate Filaments.
139. **Csaba Ráduly-Baka**, Algorithmic Solutions for Combinatorial Problems in Resource Management of Manufacturing Environments
140. **Jari Kyngäs**, Solving Challenging Real-World Scheduling Problems
141. **Arho Suominen**, Notes on Emerging Technologies
142. **József Mezei**, A Quantitative View on Fuzzy Numbers
143. **Marta Olszewska**, On the Impact of Rigorous Approaches on the Quality of Development
144. **Antti Airola**, Kernel-Based Ranking: Methods for Learning and Performace Estimation
145. **Aleksi Saarela**, Word Equations and Related Topics: Independence, Decidability and Characterizations
146. **Lasse Bergroth**, Kahden merkkijonon pisimmän yhteisen alijonon ongelma ja sen ratkaiseminen
147. **Thomas Canhao Xu**, Hardware/Software Co-Design for Multicore Architectures
148. **Tuomas Mäkilä**, Software Development Process Modeling – Developers Perspective to Contemporary Modeling Techniques
149. **Shahrokh Nikou**, Opening the Black-Box of IT Artifacts: Looking into Mobile Service Characteristics and Individual Perception
150. **Alessandro Buoni**, Fraud Detection in the Banking Sector: A Multi-Agent Approach
151. **Mats Neovius**, Trustworthy Context Dependency in Ubiquitous Systems
152. **Fredrik Degerlund**, Scheduling of Guarded Command Based Models
153. **Amir-Mohammad Rahmani-Sane**, Exploration and Design of Power-Efficient Networked Many-Core Systems
154. **Ville Rantala**, On Dynamic Monitoring Methods for Networks-on-Chip
155. **Mikko Pelto**, On Identifying and Locating-Dominating Codes in the Infinite King Grid
156. **Anton Tarasyuk**, Formal Development and Quantitative Verification of Dependable Systems
157. **Muhammad Mohsin Saleemi**, Towards Combining Interactive Mobile TV and Smart Spaces: Architectures, Tools and Application Development
158. **Tommi J. M. Lehtinen**, Numbers and Languages
159. **Peter Sarlin**, Mapping Financial Stability
160. **Alexander Wei Yin**, On Energy Efficient Computing Platforms
161. **Mikołaj Olszewski**, Scaling Up Stepwise Feature Introduction to Construction of Large Software Systems
162. **Maryam Kamali**, Reusable Formal Architectures for Networked Systems
163. **Zhiyuan Yao**, Visual Customer Segmentation and Behavior Analysis – A SOM-Based Approach
164. **Timo Jolivet**, Combinatorics of Pisot Substitutions
165. **Rajeev Kumar Kanth**, Analysis and Life Cycle Assessment of Printed Antennas for Sustainable Wireless Systems
166. **Khalid Latif**, Design Space Exploration for MPSoC Architectures

167. **Bo Yang**, Towards Optimal Application Mapping for Energy-Efficient Many-Core Platforms
168. **Ali Hanzala Khan**, Consistency of UML Based Designs Using Ontology Reasoners
169. **Sonja Leskinen**, m-Equine: IS Support for the Horse Industry
170. **Fareed Ahmed Jokhio**, Video Transcoding in a Distributed Cloud Computing Environment
171. **Moazzam Fareed Niazi**, A Model-Based Development and Verification Framework for Distributed System-on-Chip Architecture
172. **Mari Huova**, Combinatorics on Words: New Aspects on Avoidability, Defect Effect, Equations and Palindromes
173. **Ville Timonen**, Scalable Algorithms for Height Field Illumination
174. **Henri Korvela**, Virtual Communities – A Virtual Treasure Trove for End-User Developers
175. **Kameswar Rao Vaddina**, Thermal-Aware Networked Many-Core Systems
176. **Janne Lahtiranta**, New and Emerging Challenges of the ICT-Mediated Health and Well-Being Services
177. **Irum Rauf**, Design and Validation of Stateful Composite RESTful Web Services
178. **Jari Björne**, Biomedical Event Extraction with Machine Learning
179. **Katri Haverinen**, Natural Language Processing Resources for Finnish: Corpus Development in the General and Clinical Domains
180. **Ville Salo**, Subshifts with Simple Cellular Automata
181. **Johan Ersfolk**, Scheduling Dynamic Dataflow Graphs
182. **Hongyan Liu**, On Advancing Business Intelligence in the Electricity Retail Market
183. **Adnan Ashraf**, Cost-Efficient Virtual Machine Management: Provisioning, Admission Control, and Consolidation
184. **Muhammad Nazrul Islam**, Design and Evaluation of Web Interface Signs to Improve Web Usability: A Semiotic Framework
185. **Johannes Tuikkala**, Algorithmic Techniques in Gene Expression Processing: From Imputation to Visualization
186. **Natalia Díaz Rodríguez**, Semantic and Fuzzy Modelling for Human Behaviour Recognition in Smart Spaces. A Case Study on Ambient Assisted Living
187. **Mikko Pänkäälä**, Potential and Challenges of Analog Reconfigurable Computation in Modern and Future CMOS

Turku
Centre *for*
Computer
Science

Joukahaisenkatu 3-5 B, 20520 Turku, Finland | www. tucs.fi

University of Turku
Faculty of Mathematics and Natural Sciences

- Department of Information Technology
- Department of Mathematics and Statistics

Turku School of Economics

- Institute of Information Systems Science

Åbo Akademi University
Division for Natural Sciences and Technology

- Department of Information Technologies

ISBN 978-952-12-3141-4
ISSN 1239-1883

Mikko Pänkäälä

Potential and Challenges of Analog Reconfigurable Computation in Modern and Future CMOS

Mikko Pänkäälä

Potential and Challenges of Analog Reconfigurable Computation in Modern and Future CMOS